Contextual language understanding Thoughts on Machine Learning in Natural Language Processing

Benoit Favre

HDR: Contextual language understanding

Thoughts on Machine Learning in Natural Language Processing

Benoit Favre

January 1, 2020

Foreword

This document is a *habilitation à diriger des recherches (HDR)* thesis. It is organized in two parts: The first part presents a reflection on my work and the state of the Natural Language Processing community; The second part is an overview of my activity, including a detailed CV, a summary of the work of the PhD students I contributed to advising, and a list of my personal publications. Self-citations are postfixed with † and listed in Chapter 9, while external references are listed in the bibliography at the end of the document. Each contribution chapter ends with a section listing PhD student work related to that chapter.

Abstract

Natural language is pervasive in a society of information and communication. Processing it automatically, be it in the form of analysis or generation, is at the center of many artificial intelligence applications. In the last decades, the natural language processing (NLP) community has slowly adopted machine learning, and in particular more recently deep learning, as a major component of its research methodology. NLP tasks are typically expressed as structured classification problems, for which systems can be devised by finding parameters that minimize a cost function on a previously hand-labeled training corpus.

This document explores the current state of the research questions addressed by the NLP community through three interwinded challenges: taming modeling assumptions, ensuring generalization properties and dispensing adequate methodology.

Modeling assumptions are often seen from a statistical point of view, such as the assumption that training samples shall be independently distributed, yet NLP assumes other kinds of dependency assumptions which impact system performance. Input representations, and in particular the scope of input features used to characterize a decision, may need to be reconsidered when processing rich linguistic phenomena. Structured predictions for which decisions are interdependent are tackled through a compromise between rich decoding schemes with low modeling power, and suboptimal decoding with richer models. Linguistic descriptions have lead to modular approaches resulting in processing chains hindered by cascading errors, an approach questioned by recent end-to-end and multitask training.

The second challenge is performance generalization, meaning that systems shall not collapse under conditions differing from the training distribution. Generalizing NLP systems across genres, such as from news to social media, and modality, from text to speech, requires accounting for the distributional and representational shift between them. In addition, recent development of common annotation schemes over a range of languages, and the resulting availability of multilingual training data allows to explore creating systems that can process novel languages for which they have not received full supervision.

The third challenge is methodological, exemplified through problems with shared tasks and lack of user involvement in current evaluation metrics. Shared tasks, a tool used to foster progress through independent evaluation of systems developed by competing scientific institutions, are both beneficial and detrimental due to over-engineering and bias effects. Besides shared tasks, the loss functions encouraged by machine learning for training NLP systems are often generic and do not account for the context in which they are used. Developing user-oriented evaluations is a research direction for improving this involvement.

Contents

Part I

Contextual Language Understanding

Chapter 1

Introduction

In order to make machines significantly easier to use, it has been proposed (to try) to design machines that we could instruct in our native tongues. This would, admittedly, make the machines much more complicated, but, it was argued, by letting the machine carry a larger share of the burden, life would become easier for us.

Edsger W. Dijkstra, 1979

Natural Language Processing (NLP) is a field of computer science which aims at studying how human language in all its forms can be processed and generated by computers. Originating from a multidisciplinary community, it draws from the fields of mathematics, philosophy of language, linguistics, psycholinguistics, cognitive sciences, social sciences, neuroscience, in order to model and simulate human behavior regarding natural language.

A definition of the word *language*, given in (Crystal et al. 2018), could be "*a system of conventional spoken, manual, or written symbols by means of which human beings, as members of a social group and participants in its culture, express themselves. The functions of language include communication, the expression of identity, play, imaginative expression, and emotional release.*" This definition actually restricts itself to natural language, by opposition to formal languages which are mathematical constructs made of sequences of abstract objects with interesting regularities, in which fall for instance programming languages. Even though formal languages are often used to describe some aspects of natural languages, the 'natural' modifier has been included because of the complexity of the observed phenomenon of human-human communication that may be beyond the reach of simple formal languages. The debate between formalists and naturalists probably originated in Noam Chomsky's interest for the source of the phenomenon of language in the mind (embodied by the study of formal languages) and relative lack of interest for how this phenomenon translates into actual linguistic instances, corrupted by actuators and communication channel issues. Yet, corpus-based linguistics and the ensuing empirical evaluation of natural language processing systems have led to very successful applications that changed our everyday life, and that would probably not have been possible if the community did not focus on natural language at some point.

The previously mentioned definition of language also imposes interesting restrictions on the kind of symbolic systems used for conveying language, and the fact that language is a human-centric concept. While spoken language is the means of choice for human communication and the first language a baby learns, written language has been the favorite durable means of communication because it could be painted, written, printed, and sign language even if most developed in disabled communities, is a recognized complement to spoken language and one of the first languages infants can learn. A first

question entailed by this restriction is whether language can be expressed in other modalities. For instance, music is often recognized to have a lot of properties of language (Rohrmeier et al. 2015), and therefore should it be considered as a natural language? Another question is whether the symbols upon which language relies need to be discrete or could be of another form. The space of colors is inherently continuous, and language has trouble mapping it with symbols; paralinguistic information such as emotions also do not map well to a symbolic space; sign language has many non-symbolic constructs based on location or intensity of gestures. Maybe the underlying question is whether thinking is symbolic and, if not, how it maps to a symbolic language (Lupyan 2016). The second restriction is the limitation to human communication. It has been shown multiple times that animals use some form of language that have a lot in common with human language, and it is also interesting to question whether natural language processing should be limited to human language (and there are several current efforts to build dog-human translation devices, with various degrees of success).

The general public perception of natural language processing is that since language manipulation is so easy for humans, it should be as straightforward for machines. Automatic speech recognition, the generation of a transcript from what was said in an audio recording, is a typical example of an intuitively easy yet extremely difficult task. The first problem is the recognition of phonemes from the acoustic signal. Phonemes are the basic unit of sound that can be produced by the vocal conduct, such as vowels (sustained frequencies produced by driving air through the vocal folds) or consonants (produced for example by fast tongue and lip movements). Recognizing phonemes is challenging because different morphologies of the vocal conduct lead to speaker-level variability of the frequencies and timing that characterize a phoneme. It is also challenging because the vocal conduct is a continuously moving organ which results in slow transitions between the stable states of phonemes. In addition to those challenges, the acoustic channel (reverberation, distance to the microphone, type of microphone, presence of other sources of noise) may be corrupted leading to uncertainties in the recognition process. Even if the phoneme sequence were easy to reconstruct, mapping phonemes to lexical units is difficult because of ambiguity. Given a phoneme sequence, there are typically hundred thousand sequences of words that map to that sequence among which most are nonsense but some make perfect sense out of context (a good source of puns). A speech recognition system has to guess the actual sequence, which often implies understanding the intent of the original speaker. Automatically understanding such natural language, called Natural Language Understanding or NLU, is also difficult because of ambiguity (a single word might have multiple senses), because of the use of references which require to account for a broad context (for instance relying on general knowledge), and because the target of what "understanding" means (and in general what semantics is) is ill-defined and often an open research problem.

Natural language processing includes some of the most difficult problems of the field of artificial intelligence because they often require world knowledge and general reasoning (therefore, it is listed as one of the IA-complete problems, possibly requiring general Artificial Intelligence), yet many successful applications have been striving over the years. These applications rely on mimicking how humans manipulate language in a specific context or domain instead of completely simulating an intelligent system. General purpose dialog agents, also known as "chatbots", are a good example of how humans can be deceived in thinking that they are talking to an intelligent machine. The Turing test, long thought as a good test of achieving artificial intelligence, consists in blindly matching human judges with dialog agents and other humans, and measuring whether the judges can discriminate between humans and computer programs. The ELIZA chatbot (Weizenbaum 1976) or contestants to the Loebner Prize competition (Stephens 2004) are dialog systems which rely on conversational tricks in order to evade difficult questions (such as invoking boredom, switching topics, etc.) Machine Translation is another example which uses recurrent statistical properties of aligned corpora across languages in order to mash-up good-quality translations. One fundamental question of NLP is whether we should continue to make mimicries, in the same way the aeronautics community has built planes instead of simulating bird flight, or if we can overcome the difficulties of simulating cognitive functions of the brain.

While early natural language processing systems relied on formal models (the introspective approach), the community has slowly evolved towards using statistical models of language to eventually

relying on machine learning for a broad range of NLP tasks. This evolution probably began with the introduction of corpora of naturally occurring language phenomena for evaluating the quality of NLP systems. This was a step towards more ecological evaluation of language technology. As corpus size grew, it started to be possible to tune system hyper-parameters to maximize performance, and finally to use a subset of the annotated data to train machine learning algorithms. Machine learning consists in creating computer programs that can learn from experience without being explicitly designed to solve a particular task. The inference problem is often formalized as generating the output (class label, symbolic structure or set of real values) from a parametrizable function of the input (features extracted from observations), while the learning problem consists in finding the parameters of the function that best generate the correct labeling of a corpus. The approach developed in the NLP community by first replacing hand-designed rule-based systems with automatically mined rules from a superset of possible rules, which made the success, for example, of the Brill part-of-speech tagger (Brill 1992). Later, researchers developed statistical approaches leveraging independence assumptions to decompose joint probabilistic models of the labels to predict and observations, and computed the probability of events through a frequentist maximum likelihood estimation of discrete distributions. These approaches lead to success in automatic speech recognition (Huang et al. 1990), language modeling or machine translation (Koehn 2009). Another landmark approach was maximum entropy (Berger et al. 1996) and conditional random fields (Lafferty et al. 2001) which extend the frequentist statistical approach to a log-linear modeling of distributions decomposed over features of the input. The NLP community then followed more closely the advances in the machine learning community. More classes of models have been made possible to explore by formalizing the learning problem as a loss function minimization problem, and directly minimizing the end-task errors (such as approximations of BLEU in machine translation). While most prominent models were linear, nonlinear models have long been explored with decision trees, support vector machines or neural networks. Finally, recent advances in deep learning have promoted neural networks to the dominant approach for a lot of NLP problems, such as speech recognition, machine translation, etc. In major conferences and journals of the domain (Computational Linguistics, ACL, EMNLP, EACL, NAACL, COLING, etc.), the percentage of papers with titles related to machine learning has tripled from 2010 to 2018, going from around 10% to more than 30% (lower bound on actual topical content; see Table 1.1). Clearly, machine learning is establishing itself as a very strong component of the field. This observation makes one wonder whether this trend will last, or whether machine learning will eventually fade and be replaced.

Year	**Matches**	**Total**	**%**
1965-2009	1,494	20,671	7.2
2010	269	2,666	10.1
2011	207	1,856	11.2
2012	279	3,047	9.2
2013	298	2,540	11.7
2014	371	3,297	11.3
2015	464	2,587	17.9
2016	806	3,722	21.7
2017	970	3,144	30.9
2018	1,468	4,335	33.9
mid-2019	379	1,120	33.8

Table 1.1: Number of article titles of the major conferences and journals of the domain indexed in the ACL Anthology (CL, ACL, EMNLP, EACL, NAACL, COLING, etc.) that match typical machine learning words (*neural, learning, deep, training, embedding, network, end-to-end, attention, lstm, ensemble, adversarial, bayesian, supervised, margin, support, loss*) from 1965 to mid-2019 with of focus on recent years. Data collected from `https://aclanthology.info`.

Machine learning and the whole ecosystem it generates (data collection and annotation, system

development, model choice and parameter tuning, evaluation protocols and shared tasks) is favored by the community for the following reasons. First, it is considered as a step towards ecological evaluation through the use of collected naturally occurring language data. It also helps building a repeatable experimental setup because machine learning algorithms, trained with the same corpus and given the same initial parameters, lead to the same performance results on the test set. One advantage of machine learning that is a dividing argument in the community is that associated models are often linguistic-theory neural and by being able to learn relevant features from the data, require less linguistic expertise on the target problems. This aspect also allows to capture regularities that are not apparent in the data, and therefore often yields superior performance in scientific competitions. Another benefit is that machine learning models can often account for very large datasets and are relatively efficient in terms of processing. An additional reason is that some aspects of machine learning imitate the learning process of the brain, allowing computers to learn language in the same way children learn language. This last reason is debatable since it is not settled how much of the implementation of language in the brain is the result of evolution, and how much is the result of social interaction.

While machine learning may seem like a definitive answer to a very broad range of problems, it also exhibits some limitations which have been recognized by various communities. The main disadvantage is that machine learning requires much more supervision than humans for being able to obtain reasonable performance (and even sometimes super-human performance) in solving a particular class of problems. There have been many efforts to increase data efficiency, such as active learning techniques (Settles 2012) which consist in iteratively training a system and selecting which examples to annotate in order to maximize the performance gain of the system. Another trend of research is that of meta-learning in which machine learning systems are faced with numerous different tasks so that they get a chance at capturing faster the specifics of a novel task (Finn et al. 2017). In the computer vision community, it has been shown multiple times that it is easy to add noise to images that perceptually belong to a category unambiguously while a trained system gives a different category with very high confidence. Attacking machine learning systems with adversarial examples is an old topic as evidenced in the speaker verification community with voice transformations that allow spoofing an identity (Matrouf et al. 2006), or by more recent techniques for building physical objects that can consistently fool machine learning systems while being recognized by humans (Kurakin et al. 2016). In order to contain the effects of such attacks, but also required by certain application domains such as medicine, recent efforts in machine learning have focused on making systems which are interpretable at the model level (what did the model learn, how did it learn?) and at the decision level (why did the mode make a given decision?). It is often stated that deep neural networks are not interpretable, but one can remark that linear models are not much more interpretable than their deeper peers (Lipton 2016).

Given the attraction of the NLP community for machine learning in face of the benefits and drawbacks it carries, it is reasonable to question whether those specificities are important for the NLP field, and try to shape what compromises the community is making by focusing on a single approach. This document presents a set of reflections based on my experience at the saddle point between the fields of natural language processing and machine learning, and illustrated and supported by arguments based on the research I have conduced since my MSc thesis. Its aim is to introduce elements that can help us move towards answering the following questions:

- What is the impact of assumptions typically associated with machine learning on current NLP research?
- Are models trained through machine learning able to generalize well on natural language processing tasks?
- What are the limits of methodological practices in the natural language processing community in the context of machine learning?
- On what problems should the NLP community focus now that reliance on machine learning has matured?

This document does not give a definitive answer to those questions. But it shows directions which might lead to a better understanding of a range of NLP problems. Those directions are based on the idea of extending the *context* in which problems are tackled, by pushing the envelope of formalization, modeling, implementation and methodology associated with them. This approach is fundamental to *Contextual Language Understanding*.

This part of the document is organized as follows. Chapter 2 gives a broad overview of major natural language processing techniques with a focus on recent deep learning approaches. Chapter 3 is an attempt at measuring the impact of assumptions on NLP system performance. Chapter 4 tries to determine up to what point generalization is limited in current NLP approaches. Chapter 5 addresses some of the methodological concerns with current practices in empirical research in the community. The last chapter of this part gives a general conclusion and lists a few prospects in the field.

Chapter 2

Modern Natural Language Processing

Many introductions to natural language processing, encompassing historical or recent approaches, can be found in the literature. The objective of this chapter is not to build a comprehensive survey of existing problems and methods but rather to focus on select landmarks that are relevant to the next chapters. In particular, it only covers a few supervised machine learning techniques popular within the community for a large number of NLP tasks. For a more in-depth approach of NLP, the reader may refer to (Jelinek 1997; Manning et al. 1999; Martin et al. 2009; Koehn 2009; Goldberg 2016; Deng et al. 2018).

This chapter first covers the dominating methodological approach to building NLP systems based on empirical evaluation of their performance on "language-in-the-wild" corpora. Then, it outlines a broad family of NLP tasks and reviews machine learning approaches that can be exploited to tackle them.

2.1 An Empirical Approach

Empirical evaluation has become the workhorse of the natural language processing community. In terms of methodology, the main problems that need to be addressed are reproducibility (research results can be replicated), and representativeness (the experimental setting is realistic). To address both, researchers have been resorting to the notion of corpus, a set of language samples collected from actual interactions or sources, which stand as representative of the NLP problem being addressed. Corpora can be made accessible for others to replicate results, and can be collected from a variety of settings to ensure realism.

The typical approach for creating an NLP system consists in the following loop:

1. define the **task**, be it part-of-speech tagging, named entity recognition, machine translation, or summarization;

2. gather **corpora** that encompasses a range of sources, genres, styles to be representative of the applications you have in mind;

3. write a precise and coherent annotation **guide**, and define a meaningful evaluation criterion;

4. **annotate** the corpora with task labels using informed or naive annotators to build a gold standard;

5. create a **system** that will predict labels from raw data;

6. **evaluate** the system output on held-out data;
7. loop to step (1) to refine the approach – only a subset of steps might be refined (for instance, it is common to alternate between improving a system and evaluating it).

This loop, which could be called TCGASE (for Task, Corpus, Guide, Annotation, System, Evaluation), is essential not only to current NLP research, but also to system engineering in the industry as quality control. It is suitable for creating machine-learning based systems as it provides an environment with labeled data and a criterion to optimize for. In the following sections, we detail a bit more the notions of **task**, **system** and **evaluation metrics**.

2.2 Tasks

Traditionally, NLP tasks are organized according to the analysis-synthesis dichotomy, as well as the means-and-end dichotomy which considers intrinsic tasks (which are steps towards an end) and extrinsic tasks (which can be involved in end-user applications). However, the boundary between those categories is very loose as the same models can often be used for analysis and synthesis, and a lot of intrinsic tasks can be exposed in applications.

Another way of looking at NLP tasks is according to processing levels: meta, syntactic, semantic, discourse and pragmatic levels. Automatic processing at those levels is typically performed sequentially, from the lower to the higher level. A non-exhaustive list of tasks could be described as follows.

- The **Document level** operates on meta-descriptors of the linguistic content:
  - Language and code identification and segmentation: finding what language the current document contains, some documents containing multiple languages, or sequences of words using different writing systems, such as the inclusion of English words in Chinese, or mathematical symbols in an article;
  - Author identification and trait classification: characterizing the style of an author to devise her identity or traits such as social category, age, etc.;
  - Document structure analysis: determining where structural elements such as section headers, paragraphs, lists, figures, etc. are.
- The **Syntactic level** corresponds to structural elements that are linked to the function of linguistic elements:
  - Sentence splitting, punctuation prediction: finding where sentences start and end, which can be difficult in spontaneous spoken or textual content;
  - Word segmentation, tokenization, multiword expression detection: identifying words and associated lemmas in the stream of characters which might not be separated by spaces or punctuation such as in Sino-Tibetan languages;
  - Part-of-speech tagging, morphological analysis: determining the syntactic category of words, as well as traits such as gender or number which can be determined from sequences of characters within words;
  - Syntactic chunking and parsing: finding the latent structure that links words together outlining the function of groups of words under a grammar theory.
- The **Semantic level** aims at analyzing the meaning of linguistic constructs:
  - Named entity recognition, linking: identifying sequences of words which correspond to real-world entities and matching them in a database of existing entities;
  - Word-sense disambiguation: recognizing the meaning of each lexical unit among a catalog of possible meanings;

  - Topic classification and segmentation: finding the topic or hierarchy of topic segments of a document;
  - Semantic parsing: constructing the latent structure of the sense of a sentence from the meaning of individual words according to one of the semantic representation theory.

- The **Discourse level** drives how discourse and interaction are organized
  - Coreference resolution: resolving pronouns and references to aforementioned entities or external entities;
  - Dialogic parsing: analyzing the construction of a dialog or a multiparticipant conversation in terms of dialog acts, question-answer pairs, etc.;
  - Discourse parsing: determining how arguments are constructed and related.

- The **Pragmatic level** consists in the integration of meaning units in the context in which they are produced, often with an applicative end in mind.
  - Sentiment analysis and opinion mining: determine the stance or emotional state of an author, speaker or character regarding a topic;
  - Knowledge representation: build an ontology or formal representation of knowledge acquired from a text or during a communication effort;
  - Reasoning / integration: reason about what to do next given the current state of system and linguistic input (dialog systems, robots, etc.).

It is apparent that some of those task do not completely fit the category they are attached to, and that some may benefit from the output of a number of other tasks. This description towards the analysis side of NLP also holds (in reverse level order) for the generative side which aims at producing linguistic constructs given abstract representations.

Natural language processing has generated a number of applications that span the whole industry of human-human and human-machine interactions, such as machine translation, summarization, speech recognition or dialog systems. Those applications might be treated as compositions of finer-grained NLP tasks, or tasks of their own.

Most of those tasks can be cast as one of five problems:

- **Labeling**: predict an output label from a set of possible labels (a classification task), often extended to sequence or tree labeling because the label for one element depends on neighboring elements. Word sense disambiguation or part-of-speech tagging belong to that category.

- **Segmentation**: predict segment boundaries for the input, as a partition of disjoint, possibly overlapping segments. Syntactic chunking or named entity recognition are instances of this task. A non-overlapping segmentation task can be cast as a sequence labeling problem by using the popular begin-inside-outside (BIO) encoding which appends a B to the first item of a segment, an I to all items inside it and an O for items which are not in a segment[1].

- **Regression**: predict a value in a continuous space, such as for sentiment valency assessment, predicting the quantity of silence required for a dialog system to take the floor, speech synthesis (generation of a sequence of audio samples), or predicting customer satisfaction on Likert scale. When the nature of the target is not natively continuous, the problem is often mapped to (ordered) labeling.

- **Relation detection**: predict that two elements are in relation, and label that relation. Dependency parsing is cast as relation detection, but so can be coreference resolution. Some constraints can be added to the task as in dependency parsing where a word can only have one governor, which ensures that the created structure is a tree.

[1] There exist a range of representations of segments as word-level classification tasks (Konkol et al. 2015) and explicit prediction of segment boundaries (Gillick et al. 2016).

- **Generation**: predict a label which decomposes as a sequence of items (generally words). It is the case of summarization or machine translation for which a sequence of words must be generated.

The advent of deep learning has greatly extended the expressivity of machine learning systems, blurring this categorization[2].

2.3 Systems

Given this description of natural language tasks, it would be interesting to be able to reuse similar methods for dealing with different problems. The mainstream approach is to use machine learning to solve those problems.

NLP problems are often treated as classification of an input $x \in \mathcal{X}$ as a label $y \in \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are respectively the set of all possible inputs and labels. When $\mathcal{X}$ or $\mathcal{Y}$ are large, x and y are decomposed in smaller units, such as words and part-of-speech tags of a sentence. The inference problem consists in predicting y given x, with a parameterized function f_θ.

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} f_\theta(x) \tag{2.1}$$

Training consists in finding the parameters θ that minimize an empirical loss on never seen data (for example the number of mislabeled instances), which is approximated by minimizing a loss function L on a training corpus $\mathcal{C}$, which is called empirical risk minimization (ERM).

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{(x,y) \in \mathcal{C}} L(y, f_\theta(x)) \tag{2.2}$$

In the following sections, we overview a few "shallow" machine learning models and "deep" neural networks that are often used in NLP systems.

2.3.1 Standard models

Perceptron The perceptron algorithm is one of the most straightforward linear model for machine learning. If x is a feature vector, W a weight matrix of size the number of features times the number of possible labels, y is a score vector with a value for each label (the highest scoring label being the predicted one). Prediction under that model is performed as:

$$y = Wx \tag{2.3}$$

Training is achieved by stochastic gradient descent by repeatedly sampling examples from the training set, performing prediction, and adjusting the weights according to whether the correct label was predicted or not. The gradient is discretized so that -1 is added to the weights of the incorrectly highest scoring label, and 1 is added to the weights of the gold label that should have been predicted, in a maximum margin fashion. A very good extension is the averaged perceptron which consists in saving the weight matrix after each update, and eventually averaging all the weight matrices to produce the final model (Collins 2002). It can be computed efficiently by observing that the average model can be factorized according to each of the updates.

Structured prediction problems can be tackled by decomposing y in substructures and using an inference procedure, such as the Viterbi Algorithm for chains of factors, and then adjust the weights of the features linked to the substructures. The size of W depends on the number of features times the number of values the substructures can take (for a bigram tagger, they would be the number of possible tag bigrams) which tends to grow quickly in practice. Sparse representations are often used for both x and W.

[2]Systems can be set up to solve hybrids of those problems although the various losses they are trained for can be broadly mapped to them.

The perceptron algorithm has been very popular in the NLP community for its simplicity and speed, while achieving very good results in tagging and parsing tasks at a fraction of the cost of other models (Collins 2002; McDonald et al. 2005a). A number of variants of the training procedure have been proposed, such as Mira (Crammer et al. 2006) or Adagrad (Singer 2010), without leading to systematic improvements compared to the averaged perceptron.

Maximum entropy and conditional random fields Maximum entropy models are log-linear probability models which can be expressed as:

$$p_\theta(y|x) = \frac{1}{Z(x)} \exp\left(\sum_k \theta_k f_k(x,y)\right) \tag{2.4}$$

$$Z(x) = \sum_{y' \in \mathcal{Y}} \exp\left(\sum_k \theta_k f_k(x,y')\right) \tag{2.5}$$

where θ is a vector of trainable parameters and $f(x,y)$ is a binary feature vector for the label y of x. $Z(x)$ is a normalization factor which makes the probabilities over labels y sum to 1.

Log-linear models can be trained by maximizing the log likelihood of the training data, with an optional regularization penalty ($\mathcal{L}_{\text{reg}}(\theta)$, with λ regulating the quantity of regularization, determined on a development set).

$$\mathcal{L}(\theta) = \sum_i \log p_\theta(y^{(i)}|x^{(i)}) \tag{2.6}$$

$$\mathcal{L}_{\text{reg}}(\theta) = \sum_i \log p_\theta(y^{(i)}|x^{(i)}) - \frac{\lambda}{2}\sum_j \theta_j^2 \tag{2.7}$$

$\mathcal{L}(\theta)$ (and $\mathcal{L}_{\text{reg}}(\theta)$) is convex and can be maximized with any off-the-shelf algorithms such as gradient ascent or LBFGS (Liu et al. 1989). Closed-form derivation of the gradients of $\mathcal{L}$ with respect to θ can be found in (Collins 2005).

Conditional Random Fields (CRF) are maximum entropy models applied to labels y decomposable over a graph of factors. The basic idea is that an instance consists in a set of slots T which have to be labeled with atomic labels $y_t \in \mathcal{Y}_T$ so that $y = \{y_t\}_T$. Then, we can consider G_T a graph over T and factors as connected components, or cliques, of G_T. Intuitively, the factors are subsets of atomic predictions in T that represent dependent phenomena. Given that, a CRF is a log-linear model where binary features are extracted for each factor, and associated with trainable weights. Since cliques may share slots, the inference problem (finding the highest scoring labeling of the graph) must make sure that factors agree on the atomic label chosen for a given slot.

In NLP, the most commonly used form of CRF model is first-order linear chain CRFs. If $x = x_1 \dots x_T$ is a sequence of observations, and $y = y_1 \dots y_T$ is a sequence of labels, then $p_\theta(y|x)$ can be expressed as:

$$p_\theta(y|x) = \frac{1}{Z(x)} \prod_{t=2}^{T} \exp\left(\sum_k \theta_k f_k(x, y_t, y_{t-1})\right) \tag{2.8}$$

$$Z(x) = \sum_{y'} \prod_{t=2}^{T} \exp\left(\sum_k \theta_k f_k(x, y'_t, y'_{t-1})\right) \tag{2.9}$$

where factors correspond to bigram of labels in the instance, and $Z(x)$ is summed over all possible labeling of the input. This extends to higher order (larger n-grams), and other types of structure, such as trees or graphs.

Training is similar to that of maximum entropy models, except that inference is performed with the Viterbi algorithm (for sequences) which finds the maximum probability labeling of an instance, and the forward-backward algorithm which can compute the $Z(x)$ normalization factor efficiently.

CRFs have been very successful in tagging tasks, such as part-of-speech tagging or named entity recognition (Toutanova et al. 2003).

Support vector machines A kernel is a similarity function between two instances that satisfies the Mercier property. In particular, it has the following property:

$$K(x_i, x_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle \tag{2.10}$$

which means that the similarity may be computed in a projection space ($\phi(x)$) as a scalar product $\langle \cdot \rangle$. The main idea of Support Vector Machines is to find a separator hyper-plane in the projection space instead of the original representation space of the data, where the projection space is of potentially larger dimension than the original to ensure that the classes are separable (known as the kernel trick).

The classifier is trained by finding a separator that classifies examples with a maximum margin criterion, by using slack variables ξ_i which correspond to the penetration of an example in the space of the opposite class. If $y_i \in \{-1, 1\}$ is the label of example x_i, the C-SVC formulation (Cortes et al. 1995) of SVMs with L2 regularization is trained as:

$$\min_{w,b,\xi} \quad C \sum_i \xi_i + \frac{1}{2} \sum_k w_k^2 \tag{2.11}$$

$$\text{subject to} \quad y_i(w^\top \phi(x_i) + b) \leq 1 - \xi_i \quad \forall i \tag{2.12}$$

$$\xi_i \leq 0, \quad \forall i \tag{2.13}$$

where w is a weight vector, b a bias term, C a hyper-parameter that must be set on a development set, $\phi(x_i)$ is the projection and ξ_i is the slack for example i. The dual of this objective is minimized with quadratic programming techniques.

Inference of the trained classifier is defined as:

$$y = \text{sgn}\left(\sum_i y_i \alpha_i K(x_i, x) + b \right) \tag{2.14}$$

where (x_i, y_i) are training examples, α_i is obtained during training from the dual problem and K is the kernel.

SVMs have been very popular in the machine learning community because of the associated theoretical results. They have been used for a range of NLP problems such as sentiment analysis (Vinodhini et al. 2012).

Boosting Boosting is an ensembling method which consists in building a linear combination of weak classifiers in order to obtain a stronger classifier. Many schemes have been proposed for learning such combination, but Adaboost (Schapire et al. 1999) was one of the most successful. It consists in iteratively selecting a set of classifiers from the ensemble and weighing them in order to maximize performance. It alternates between weighting the examples from the training set so that erroneous predictions according to the combination have more weight, and then selecting the weak learner which contributes best to the weighted loss. BoosTexter was a very successful implementation of Adaboost on decision stumps, which are one-level decision trees, such as decision thresholds for real-valued features, and presence detectors for textual features (Schapire et al. 2000).

The decision function for an Adaboost classifier after T rounds of training is of the form:

$$y = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{2.15}$$

where $h_t \in \mathcal{H}$ is a weak learner selected at round t and α_t is the scalar weight associated with the learner at that round. Note that the same learner may be selected multiple times.

The training algorithm for Adaboost Real.MH is as follows[3]. Let X be a matrix of m examples, and Y the associated one-hot encoded matrix for k labels. We define D a distribution over examples and labels. $h \in \mathcal{H}$ are functions (weak learners) that return a confidence matrix over their predictions for the labels for all examples. At each iteration, the weak learner h_t which minimizes the function Z is selected. The distribution D is then updated according of the weighted errors of the classifier.

$$Z(D, \alpha, h) = \text{sum}(D \odot e^{-\alpha Y \odot h(X)}) \tag{2.16}$$

$$D_1 = \frac{1}{mk} \tag{2.17}$$

$$\alpha_t = 1 \qquad \text{(Boostexter already accounts for } \alpha \text{ in } h) \tag{2.18}$$

$$h_t = \underset{h \in \mathcal{H}}{\text{argmin}}\, Z(D_t, \alpha_t, h) \tag{2.19}$$

$$D_{t+1} = \frac{D_t \odot e^{-\alpha_t Y \odot h_t(X)}}{Z(D_t, \alpha_t, h_t)} \tag{2.20}$$

where sum$(\cdot)$ returns the sum of all elements of a matrix and $\odot$ is the elementwise multiplication. In Boostexter, $\alpha_t = 1$ and $h(X)$ is defined as follows:

$$W_b^j = \text{sum}_0 \left(D_t \mathbb{1}\left[s(X) = j \wedge Y = b\right]\right) \tag{2.21}$$

$$c_j = \frac{1}{2} \ln \left(\frac{W_1^j + \varepsilon}{W_0^j + \varepsilon} \right) \tag{2.22}$$

$$h_t(X) = c_1 s(X) + c_0 (1 - s(X)) \tag{2.23}$$

where $\text{sum}_0(\cdot)$ is the sum over rows of its argument, $\mathbb{1}[\cdot]$ is the indicator function, $\varepsilon = \frac{1}{mk}$, $s(.)$ is a decision stump which returns 1 if a predicate over X is true, 0 else (typical predicates include presence of a symbolic feature, or the fact that a feature is above a threshold). c_j is the confidence of the decision stump for a given decision according to the weight distribution on the training set.

In the past, Adaboost has been quite popular[4] in the NLP and in particular the NLU communities due to its ability to seamlessly account for word n-grams, unknown features, and blend symbolic and continuous features. Gradient boosting (XGBoost) is a variant which is still very popular for its performance especially among Kaggle competition participants.

2.3.2 Deep models

As stated earlier, training a machine learning algorithm consists in finding a set of parameters that minimize a loss L between the predicted label and reference label for all training examples. This minimization process can be achieved through gradient descent techniques which require computation of the gradient with respect to the parameters θ of the loss[5] for a given training instance. Finding analytically the gradient can become difficult depending on the combined loss and inference functions. Yet a neat trick has allowed to automatically compute the gradient of deep function compositions, giving birth to deep learning.

The methods that lie under the deep learning name, are in fact quite a general class of inference functions which can be relied on to compute the loss. In particular, L is expressed as a computational graph of which composing functions need only to be partially derivable in regard of their input. Figure 2.1 shows a node of the computation graph with two inputs and one output. Using the chain

[3] The one implemented in ICSIBoost (Favre et al. 2007c).

[4] Some say that Boostexter is the first classifier you should try when approaching a new problem since it consistently yields a strong baseline, as evidenced by the success of related implementations on the MLComp platform (Liang et al. 2010).

[5] One very common loss for training classification systems is the cross-entropy loss $L_{\text{ce}}(y_t, y_p) = y_t \ln y_p + (1 - y_t) \ln(1 - y_p)$.

rule, one can compute the derivative of L in regard of each of the parameters of f as the product of the partial derivatives of each component of the computation graph in the paths that link a parameter θ_i to L. The back-propagation algorithm consists in using gradient descent to minimize L and propagate the partial derivatives along the graph from L towards the parameters.

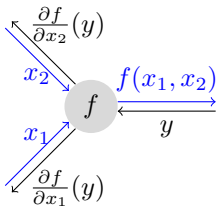

Figure 2.1: Computational graph node for a function f with two inputs x_1 and x_2 and one output $f(x_1, x_2)$. In the forward pass it computes f from its input, while in the backward pass it receives a desired output and computes the partial derivative of the input for that output.

Functions that can be expressed as computational graphs are somewhat abusively called neural networks[6] because they often rely on the basic building block of applying a linear transform to the input vector followed by an element-wise parameter-free non-linear function such as $tanh(Wx + b)$. A number of machine learning approaches can be re-expressed under the deep learning framework, by for instance using a neural network to build a representation from the input, and then feed it to the classifier. Conditional Random Fields can be trained this way (Artieres et al. 2010).

Input Representation A typical representation for categorical input such as text is a one-hot vector (or one-of-n) where all the components are zero except the one which represents the symbolic value of the input. If the input is a vector of features, then a representation can be created by concatenating the one-hot representations for each of the features. Such representations have been widely adopted with shallow models because of the induced sparsity. Since deep models do not rely on this sparsity, they can benefit from richer representations.

The problem with using words as features is that the lexicon is rich and it is very likely to encounter words at test time which are unknown from the training set. This is especially relevant for languages with a productive morphology. In addition, lexicology shows that words have a range of relationships (such as synonymy, hyperonymy, grammatical traits) which are not encoded by one-hot representations. In order to alleviate both problems, a number of techniques under the name "word embeddings" have been proposed. Most of them rely on the distributional hypothesis according to which words occurring in similar contexts tend to have similar meaning (Harris 1954).

Pioneered by techniques like LSA (Deerwester et al. 1990), the general idea is to build a low-dimension approximation of a cooccurrence matrix between words in a given window, built on a very large corpus of text. The recent model GloVe builds vector representations for words so that their dot product approximates the log of their cooccurrence (Pennington et al. 2014). The skip-gram model used in word2vec (Mikolov et al. 2013) and fasttext (Bojanowski et al. 2016), two other well-known techniques, makes sure the dot-product of words that occur in the same context is higher than that of words which occur in different contexts. Instead of words, one can use subword-units such as morphemes (Qiu et al. 2014) or supra-units such as subtrees from the dependency parse (Levy et al. 2014) in order to represent word meaning, obtaining embeddings with different properties. It has been shown that word embeddings exhibit the interesting behavior that linear transformations in the representation space encode linguistic regularities, such as plural, gender, or semantic relationship (Mikolov et al. 2013). While a number of evaluation corpora have been built to assess that property, it seems to

[6] Neural networks should probably have been called *chain-rule machines* to be more accurate.

be valid only for frequent words, and we are yet to see concrete applications which benefit from it. Recent work on building multilingual word embeddings seeks cross-lingual representations in order to build NLP systems that can be transferred from one well-resourced language to another less-resourced language (Ammar et al. 2016b). Word embeddings have flooded NLP conferences, and it is difficult to cite all the work related to their computation, evaluation and use (Li et al. 2018).

Once computed, word embeddings can be used to initialize lookup layers in neural networks, which are either fixed so that the embeddings for unknown words[7] can be used at test time, or learned with the rest of the model (fine-tuned). Embeddings can be used for other features than words, such as morphology, or even characters.

Convolutional neural networks Convolutional neural networks (CNN) originated in the computer vision community where they were developed to simulate the detection of small patterns by receptive fields[8] in the visual cortex (LeCun et al. 1998). In the context of natural language processing, they are more reminiscent of the bag-of-word hypothesis because they provide a location invariant. In fact, a typical CNN recognizes word n-grams in a word sequence.

Let $x_1 \dots x_n$ be a sequence of words, a convolutional filter at a position i and with window length l can be defined as:

$$\text{conv}_l(x, i) = \sigma(W vec(x_{i-\frac{l}{2}:i+\frac{l}{2}}) + b) \tag{2.24}$$

where $\sigma(\cdot)$ is an activation function, often ReLU (Nair et al. 2010), $vec(\cdot)$ is an operator which concatenates its input vectors in order to make a single vector, W and b are trainable parameters of the layer. The convolutional filter is then repeated for all positions i (sequence boundaries are padded with either special vectors or a continuation of the input). While this filter is known to learn a sequence of l word embeddings, multiple such filters can be used in parallel in order to learn multiple n-grams. This is achieved by increasing the number of columns of W and b.

A pooling operator is then introduced in order to provide position invariance, for instance the $\max-$pooling operator uses the location for which the activation is maximum, therefore acting as a bag of n-grams.

$$\text{pool}_{max}(x) = \max_{i=1}^{n} \text{conv}_l(x, i) \tag{2.25}$$

While variants have been proposed, $CNN(x) = pool_{max}(x)$ is one of the most popular implementations. It can learn which embedding n-grams are important in the input irrespective of their location. Because computations can be parallelized across n-gram location, the efficiency of CNNs makes them specifically suitable for processing large inputs.

Recurrent neural networks Recurrent neural networks (RNN) operate over a sequence of observations / predictions and maintain a hidden state which is updated after seeing each observation. They can be trained by unrolling them through time, and treating them as a large DNN where some weights are shared across time steps. A simple Elman recurrent neural network might be implemented as:

$$h_0 = 0 \qquad \text{initial state} \tag{2.26}$$

$$h_{t+1} = \tanh(W_r x_t + U_r h_t + b_r) \qquad \text{recurrence} \tag{2.27}$$

$$y_t = \text{softmax}(W_o h_{t+1} + b_o) \qquad \text{output} \tag{2.28}$$

Where h_0 is the initial state, x_t and h_t are the input and hidden states at time t, h_{t+1} is computed as a function of x_t and h_t and used both for the next time step and the prediction of the label y_t in the output layer.

[7] Words existing in the embedding training corpus, but not in the NLP system trained for the target task.

[8] CNNs share their weights between receptive fields, obtaining position independence. One may wonder how the visual cortex implements the idea of weight sharing.

This kind of architecture is typical of tagging tasks, but recurrent neural networks can be used for other tasks such as classification (sentiment analysis), text generation (caption generation) or transduction (machine translation, summarization, conversational agents). For classification tasks, instead of predicting one label per time step, one can input the hidden state at the end of the sequence to a decision layer which performs classification. This RNN is called an encoder. Similarly, the initial state h_0 can be replaced by a representation computed with another neural network in order to condition the predictions on a factor (such as the image for caption generation). This RNN is called a decoder. The encoded-decoder approach where an input is read, and then an output is generated is widely adopted for machine translation (Sutskever et al. 2014).

RNNs are trained by unrolling the network of a number of time steps, and back-propagation is truncated after T time steps to render the problem tractable.

Cells In order to better model long-term dependencies, a number of variants have been proposed for computing the hidden state. LSTM (long short-term memory) cells use a gating mechanism for allowing the model to learn when to memorize a value from the input and when to use it:

$$i_t = \sigma(W_i x_t + U_i h_t + b_i) \quad \text{input} \tag{2.29}$$
$$f_t = \sigma(W_f x_t + U_f h_t + b_f) \quad \text{forget} \tag{2.30}$$
$$o_t = \sigma(W_o x_t + U_o h_t + b_o) \quad \text{output} \tag{2.31}$$
$$c'_t = \tanh(W_c x_t + U_c h_t + b_c) \quad \text{cell state} \tag{2.32}$$
$$c_{t+1} = f_t \odot c_t + i_t \odot c'_t \tag{2.33}$$
$$h_{t+1} = o_t \odot \tanh(c_{t+1}) \tag{2.34}$$
$$\text{LSTM}(x_t, h_t, c_t) = (h_{t+1}, c_{t+1}) \tag{2.35}$$

An interesting aspect of LSTMs is that eq.2.33 creates a path where the gradient does not cross any non-linearity, which is supposed to reduce the gradient vanishing problem. Highway layers use the same idea to ensure the propagation of the gradient in multilayer architectures (Srivastava et al. 2015a). A number of variants of LSTMs have been proposed in order to carry fewer parameters, or model different phenomena. Gated Recurrent Units (GRU) are often used in the NLP community because on most problems they are faster than LSTMs without significant performance degradation (Chung et al. 2014).

$$z_t = \sigma(W_z x_t + U_z s_t + b_z) \quad \text{update} \tag{2.36}$$
$$r_t = \sigma(W_r x_t + U_r s_t + b_r) \quad \text{forget} \tag{2.37}$$
$$h_t = \tanh(W_h x_t + U_h(r_t \odot s_t) + b_h) \quad \text{input} \tag{2.38}$$
$$s_{t+1} = (1 - z_t) \odot h_t + z_t \odot s_t \quad \text{new state} \tag{2.39}$$
$$\text{GRU}(s_t, x_t) = s_{t+1} \tag{2.40}$$

Encoder-decoder architectures A language model can be implemented with RNNs by predicting the next word given the current word and the hidden state which encodes the history without a fixed horizon like n-grams. Such a language model can be conditioned on some arbitrary input in order that it gets biased towards generating different word sequences given different inputs. The bias could introduce additional information concatenated to word representations or as the initial hidden state (instead of setting it to 0). The additional information could be topics (Mikolov et al. 2012), a representation for an image (Xu et al. 2015) or some arbitrary data. Such a model is often called a *decoder*. Conversely, the hidden state after seeing a number of words is supposed to carry that history and can be used as a representation of its content. By dropping the RNN output layer and leveraging the hidden state after seeing the last word, one can use it as representation for categorizing the text (for sentiment analysis for example (Rouvier et al. 2016a[†])). Such an RNN is called *encoder*.

The *encoder-decoder* framework first creates a representation of the input, and then generates a sequence of words conditioned on that input. The greatest success of this idea is Neural Machine Translation (Bahdanau et al. 2014) which uses an encoder to create a representation of a sentence in the source language, and then generates the translation in the target language using a decoder. Another interesting application is learning conversational agents and question answering systems which given a representation of a question generate an answer word by word (Vinyals et al. 2015). Encoders and decoders need not be RNNs and could be any kind of representation-generating and output-generating neural networks. The point of the approach is that both are trained end-to-end so that the learned representations are tailored to the generation task (text to text, image to text, text to image, etc.).

Advanced architectures A range of architectures have been proposed to extend the speed of sequence prediction neural networks over large inputs such as speech data.

Quasi-Recurrent Neural Networks (Bradbury et al. 2016) and variants make sure that a large portion of matrix multiplications involved in computing activations of a sequential layer can be parallelized. The simplest form consists in only using a forget gate, so that the input transformation z_t can be computed for a whole sequence at once in a single matrix multiplication.

$$z_t = \tanh(W x_t + b) \tag{2.41}$$
$$f_t = \sigma(W x_t) \qquad \text{forget gate} \tag{2.42}$$
$$h_t = f_t \odot h_{t-1} + (1 - f_t) \odot z_t \tag{2.43}$$

QRNNs have been shown to perform as well as LSTMs on a few tasks requiring more layers but at a fraction of the execution time.

This idea can be extended to fully convolutional networks with a non-recurrent gating mechanism. Gated Linear Units (Dauphin et al. 2016) apply two convolutions on the input, and use the second one as gating mechanism for the first one. Again, this architecture works if multiple layers of GLU are stacked, the convolution window at a given layer helping characterize a larger context.

$$h_t = \text{conv}_l^{(1)}(x, t) \odot \sigma\left(\text{conv}_l^{(2)}(x, t)\right) \tag{2.44}$$

Another interesting approach to accessing long term contextual information is dilated convolutions (Yu et al. 2015) which operate like convolutions, but on inputs distant of a dilatation factor d. At each layer, the dilatation factor is increased in order to reach further context. This approach was successfully applied to the text-to-speech and speech-to-text tasks (Oord et al. 2016).

2.3.3 Additional design patterns

Bidirectional networks The first issue with RNNs is that even though they can potentially memorize long-distance phenomena, they are limited by their temporal causality. A straightforward extension consists in building two RNNs, one that runs from forward in time from 0 to n, and another that runs backwards from n to 0. Their output at each time step is then concatenated to produce a hidden representation which is both influenced by the past and the future.

$$h_0^{\text{f}} = h_n^{\text{b}} = 0 \tag{2.45}$$
$$h_t^{\text{f}} = \text{RNN}^{\text{f}}(h_{t-1}^{\text{f}}) \tag{2.46}$$
$$h_t^{\text{b}} = \text{RNN}^{\text{b}}(h_{t+1}^{\text{b}}) \tag{2.47}$$
$$h_t = h_t^{\text{f}} \oplus h_t^{\text{b}} \tag{2.48}$$

where $\oplus$ is the concatenation operator, $\cdot^{\text{f}}$ and $\cdot^{\text{b}}$ respectively correspond to variables linked to the forward and backward RNNs. The produced representation h_t can then be used to predict, for instance, word-level labels.

Multilayer networks In order to build more abstract representations, it is common practice to stack multiple layers of neural networks which each feeds from the representations created by underlying layers. In the vision community, it has been shown that layers closer to the input recognize simple shapes while deeper layers match more abstract concepts. In NLP, multiple layer CNNs will be able to capture combination of n-grams and multilayer RNNs can leverage phrase-level or sentence-level representations.

Multilayer architectures are riddled with the same gradient vanishing problem that RNNs suffer along the time dimension, but from layer to layer. DenseNets for instance, consist in connecting each layer to all the subsequent layers, including the output, therefore providing a shorter path for supervision propagation (Huang et al. 2016). Residual connections (He et al. 2015), consist in systematically adding the input of a layer to its output:

$$L_{i+1} = F(L_i) + L_i \qquad \text{residual connection} \tag{2.49}$$

where L_i is the output at layer i and $F(\cdot)$ is the non-linear function performed by a given layer. In this topology, there is a linear path from each layer to the output for gradients to propagate. The last class of multilayer connection is reminiscent of LSTMs as it includes a gating mechanism which choses between a layer's input and its transformed output (Srivastava et al. 2015b).

$$g_i = \sigma(WL_i + b) \qquad \text{gate} \tag{2.50}$$

$$L_{i+1} = g_i \odot F(L_i) + (1 - g_i) \odot L_i \qquad \text{highway connection} \tag{2.51}$$

where g_i is a gate vector, $\odot$ is the elementwise multiplication. Here, if $g_i = 1$, the layer output is used, while if $g_i = 0$ its input is used.

Attention Mechanisms One observation with RNNs and in particular encoder-decoder networks is that they tend to forget about specific events that occurred at the beginning of the sequence they process (Bahdanau et al. 2014). The problem is that fixed-sized representations (such as the hidden state of an RNN) cannot account for all the information available in a variable-size input. Loosely modeled after attention in human cognition, the attention mechanism in neural networks creates a global representation given a sequence of states as a weighted sum over these states. The weight distribution is parameterized on the decoder state so that the neural network can learn to focus on different input states when performing predictions.

Let e_j be an encoder state, and d_i be a decoder state, one implementation of an attention mechanism in an encoder-decoder can be defined as

$$\alpha_i = \text{softmax}_j(f_{\text{align}}(d_i, e_j)) \tag{2.52}$$

$$\text{ATTN}_i = \sum_j \alpha_{i,j} e_j \tag{2.53}$$

where f_{align} is a function which computes an alignment score between a pair of encoder and decoder states. Two broad types of alignments have been proposed: additive and multiplicative attentions (Bahdanau et al. 2014; Luong et al. 2015):

$$f^{+}_{\text{align}}(d_i, e_j) = v^\top tanh(W_1 d_i + W_2 e_j) \qquad \text{additive attention} \tag{2.54}$$

$$f^{\times}_{\text{align}}(d_i, e_j) = d_i^\top W_3 e_j \qquad \text{multiplicative attention} \tag{2.55}$$

where v is a parameter vector, W_1, W_2 and W_3 are parameter matrices.

In machine translation, attention mechanisms tend to learn an alignment between source and target words, and can be used to map unknown source words to target locations and translate them separately. Attention mechanisms are not limited to RNNs and can be applied to any representation, such as convolutions in images (Xu et al. 2015). In particular, they are used to implement memory lookups in memory networks (Sukhbaatar et al. 2015), and can even replace RNNs when stacked as multiple layers and combined with a position-encoding representation (Vaswani et al. 2017).

Ensembles Since gradient descent is not guaranteed to find the optimum of a non-linear loss function, different random initialization can lead to very different models although they have similar performance. Creating model ensembles by averaging the pseudo-probability score vectors they generate, has consistently led to large performance increase (Zhou et al. 2002a; Gong et al. 2017; Józefowicz et al. 2016) but reminds us that we are lacking insight on the true capacity of models and that empirical evidence of the superiority of a given model is relative.

Architecture Search One problem with deep learning is that there is no guarantee that a given architecture is the best for a given problem. Practitioners often rely on empirical search for the best set of hyper-parameters for a system by running many training iterations in parallel and maximizing performance on a development set. This very costly activity is a crucial component of successful systems and a typical setback for beginners. This area of research is very active and has yielded interesting results.

One can train a neural network to generate architectural parameters for another neural network. (Zoph et al. 2016) have shown that alternative gating mechanisms for LSTMs could be found by a system trained this way leading to improvements on difficult benchmarks. Other efforts look at ways of improving the efficiency of covering a large search space (Liu et al. 2018).

2.4 Evaluation

While the machine learning community has created a large set of loss functions addressing many needs, they all have in common that they are defined with the objective of facilitating the loss minimization problem. For example, losses used in deep learning should be easily decomposed in a computation graph from which a gradient can be derived. However, natural language processing problems have specifics that might not completely match those requirements. In this context, the machine learning loss is used as a proxy for the real metric used for system evaluation. Such metrics are defined by incorporating task specificities (such as the presence of multiple references as in machine translation), and relating them to the end applications (choice of class weight in a classification experiment).

Evaluation in NLP can be devised in two categories. Manual evaluation involves judges who look at each instance labeled by a system and devise a rating for it; Automatic evaluation leverages a hand-annotated corpus in order to streamline the evaluation process. A completely automatic evaluation metric, one that would not require human intervention, is still elusive because there is no sensible theoretical model to drive its definition. If such metric existed, it could be leveraged by machine learning to build better systems.

2.4.1 Manual Evaluation

Manual evaluation is necessary when a ground truth corpus cannot be easily and cheaply built. It is the case for instance in generative tasks such as machine translation or summarization, and highly subjective tasks such as emotion and sentiment classification. Irrespective of the method for manual evaluation, one fundamental issue is that humans are not consistent in their judgment. Due to the vagueness of the evaluation procedure (if it wasn't vague, it could be easily implemented as a program), judges tend to disagree among them and be inconsistent in time (due to the learning effect, the fact that judges get used to a repeated task). Therefore, it is important to use multiple judges per evaluated item, and to use techniques such as the Latin square design to limit sequential effects (Bradley 1958).

Likert Scales Likert scales are dimensions in which an instance can be evaluated according to a continuous value or more commonly discrete values in an interval. For example, the automatic summarization community has used two scales for evaluating the quality of a summary: readability/fluency and overall responsiveness. The first one assesses whether the summary is written in well-formed

English independent of its content (often subcategorized as grammaticality, non-redundancy, referential clarity, focus, and structure and coherence), while the second one assesses whether the summary covers all important content from the source documents. Both metrics are evaluated on the following scale (Dang et al. 2008):

1. Very poor
2. Poor
3. Barely acceptable
4. Good
5. Very good

Typical problems with Likert scales are the conservative decision problem where judges tend to select the middle category because of a confound between a rating and the confidence in that rating, range discrepancy problems where judges use different ranges in the given scale requiring normalization to aggregate the data, and the formulation problem where the offered category descriptions influence adversely the judges (such as having agreement categories (such as "I strongly agree") for non-subjective judgments.

System comparisons Discovering and understanding the source of system differences are important in system design. Sometimes, Likert scales cannot shed light on such differences in that it is more productive to directly compare system outputs. The most straightforward approach for system comparisons consists in displaying pairs of system outputs to judges and collecting their preference according to a set of criteria. Those pairwise comparisons could be mediated by a reference output example, as for example in summarization one would display two system-generated summaries and one reference summary as ground for the comparison. Then the question could be which system conveys more the content of the reference? Pairwise comparisons can then be aggregated by ranking operators such as Kendal rank correlation. Comparing more than two systems is possible but made cumbersome by user experience challenges (should the systems be sorted by adequacy? should only the best system be selected? how does the UI affect the produced rankings?)

Manual Matching In NLP tasks that involve composing elements (such as semantics), it is possible to create a gold standard for an instance with subitems that need to be matched to the items of the system hypothesis. While many metrics rely on automatic matching, manual matching is an option for when it is unreliable. HTER (human-targeted translation error rate) consists in manually applying edits to a translation hypothesis until it becomes one of the reference translations (Snover et al. 2006). The Pyramid method for evaluating summaries consists in first listing minimal semantic content units (SCUs) from a set of reference summaries for a given instance, and manually matching them to segments of text in a system-generated summary.

2.4.2 Automatic evaluation

In the following, $|\cdot|$ denotes the number of elements of a set. ref is a reference or gold standard for a given instance, hyp is a hypothesis or system output for that instance. The most straightforward evaluation metric is accuracy.

$$\text{acc} = \frac{|\text{correct}|}{|\text{ref}|} \tag{2.56}$$

The error rate can be devised from the accuracy metric.

$$\text{err} = \frac{|\text{errors}|}{|\text{ref}|} = 1 - \text{acc} \tag{2.57}$$

Accuracy is widely used in NLP to measure system performance on problems in which predictions are synchronous to the reference. For instance, POS tagging is evaluation with accuracy at the tag level. In dependency parsing, we use LAS (labeled attachment score) and UAS (unlabeled attachment score) metrics, which are respectively the number of correctly identified and labeled governors (LAS) and the number of correctly identified governors (UAS). For both, there is one prediction per word, so |ref| is the number of words.

Asynchronous evaluation When predictions are not synchronous between the gold standard and the system output, one can use the F-score, also known as F-measure, as a metric. It is the harmonic mean between the Recall (number of correctly predicted events from the reference) and the Precision (number of correct system predictions), and comes from the literature of information retrieval where one seeks a set of relevant documents (correct) from a large set of |ref| documents. The system retrieves |hyp| documents, and a subset of those are correct.

$$R = \frac{|\text{correct}|}{|\text{ref}|} \tag{2.58}$$

$$P = \frac{|\text{correct}|}{|\text{hyp}|} \tag{2.59}$$

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R} \tag{2.60}$$

$$F_1 = \frac{2PR}{P+R} \tag{2.61}$$

where β is an adjustment factor between the importance of recall and precision for the target application. $\beta = 1$ is used in most application domains. For example, F_1-score is used for named entity recognition because one may detect spurious entities and miss some entities, requiring to differentiate those types of errors.

Alignment Guessing which prediction is associated with which reference label is not always straightforward. In that case, one has to consider the evaluation problem as one of finding the alignment between the reference and the hypothesis which minimizes the number of errors (the same idea can be applied in order to maximize correct matches).

$$\text{metric} = \underset{\text{alignment}}{\text{argmin}}\ \text{cost}(\text{alignment}(\text{ref}, \text{hyp})) \tag{2.62}$$

An example of such metric is the Word Error Rate (WER) used in automatic speech recognition which aligns the transcript of the hypothesis with the reference through Levenshtein alignment (finding the minimum number of insertions, deletions and substitution between two sequences). The number of errors is then the sum of the numbers of insertions (ins), deletions (del) and substitions (sub).

$$\text{WER} = \frac{|\text{ins} + \text{del} + \text{sub}|}{|\text{ref}|} \tag{2.63}$$

WER can be greater than 100% if there are more words in the hypothesis than in the reference.

Extensions of WER have been proposed for machine translation where the alignment is not monotone (some constituents might be moved in the sentence) in a metric called Translation Error Rate (TER). The alignment can be performed efficiently through a beam search over the possible word displacements, also called shifts, computing at each step the Levenshtein distance between the modified strings (SNOVER et al. 2005). The METEOR metric improves over TER by using equivalence classes between words and expressions, derived from WordNet, to get better alignments between translations which do not use the same wording (Banerjee et al. 2005).

Bag-of-Matches The alignment approach assumes that it is easy to determine whether a subset of the hypothesis matches the reference. It also assumes a structural constraint: that matches are ordered, for example. Due to the fact that there is not canonical semantic representations for texts, it might be interesting to devise more lenient metrics. In generation tasks such as automatic summarization or machine translation, the community has created metrics based on bags of matches, matching small substrings from the reference irrespective of order.

First, let $\mathcal{I}$ be a set of test instances ($i \in \mathcal{I}, i = 1 \dots |\mathcal{I}|$); let $\mathcal{H}$ be a corpus of references (each instance i might have multiple references $\mathcal{R}_i$), and $\mathcal{H}$ be a set of system-generated hypotheses ($\mathcal{H}_i$ is the hypothesis for instance i). Then, we can define cm, the clipped number of matches of an element in $\mathcal{R}_i$ and $\mathcal{H}_i$. Elements are defined later but they are generally based on words.

$$\text{cm}_i(x) = \min(\text{count}(x, \mathcal{R}_i), \text{count}(x, \mathcal{H}_i)) \tag{2.64}$$

where $\text{count}(x, y)$ is the number of times x occurs in y. The clipped number of matches $cm(\cdot)$ is useful for preventing to give extra credit to words artificially added to system output.

For automatic summarization, Lin (2004) proposed to compute the overlap of word n-grams between a hypothesis summary and a set of reference summaries. ROUGE_n for n-grams of length n is defined as:

$$r_n(\mathcal{R}_i, \mathcal{H}_i) = \frac{\sum_{u \in \text{n-grams}(\mathcal{R}_i)} \text{cm}_i(u)}{\sum_{u \in \text{n-grams}(\mathcal{R}_i)} \text{count}(u, \mathcal{R}_i)} \tag{2.65}$$

$$\text{ROUGE}_n = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \text{jacknife}(\{r_n(\mathcal{R}_i \setminus j, \mathcal{H}_i), j \in R_i\}) \tag{2.66}$$

where n-grams$(\cdot)$ denotes the word n-grams of summary, $r_n(i)$ is the (clipped) n-gram recall for instance i, and jacknife$(\cdot)$ is a sampling procedure that applies jacknifing (Efron et al. 1981) to the reference sets diminished by one reference.

In practice, ROUGE also adds preprocessing to the n-grams through lowercasing, stemming and stopword-stripping of the input[9].

BLEU, BiLingual Evaluation Understudy (Papineni et al. 2002), is a similar metric for machine translation. Whereas ROUGE is a recall metric which requires that the input be clipped to a maximum length for preventing artificial inflation of scores, translation does not enforce a length constraint, and therefore BLEU is defined as a precision-based metric modified to disallow scoring inflation from adding more words to the system output. BLEU is also computed at the corpus-level instead of being a mean over instances.

$$p_n = \frac{\sum_{i \in \mathcal{I}} \sum_{u \in \text{n-grams}(H_i)} \text{cm}_i(u)}{\sum_{i \in \mathcal{I}} \sum_{u \in \text{n-grams}(H_i)} \text{count}(u, \mathcal{H}_i)} \tag{2.67}$$

$$\beta = e^{1 - \frac{|\mathcal{R}|}{|\mathcal{H}|}} \quad \text{if} \quad |\mathcal{H}| \leq |\mathcal{R}| \quad \text{else} \quad 1 \tag{2.68}$$

$$\text{BLEU}_N = \beta \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.69}$$

where p_n is the corpus-level (clipped) n-gram match precision, β is a brevity penalty based on the difference in length between the hypothesis and the reference, and BLEU_N is the metric score for n-grams of length up to N.

While similar in spirit, BLEU and ROUGE have indeed quite different implementations. Both are hard to use directly as losses in machine learning, yielding the reliance on surrogates (Zhukov et al. 2017) or reinforcement learning (Ryang et al. 2012).

[9] The ROUGE 1.5.5 implementation also clips matches to the number of references (preventing extra scoring from redundant information), and uses jacknifing (a bootstrapping method which computes the actual score as the mean of scores from all the sets of references constructed with one reference ablated) for generating more reliable measures.

Machine learning for evaluation As shared tasks have flourished in various domains of NLP, they have often been associated with accurate manual evaluations (such as adequacy/fluency for machine translation) and cheap, less accurate metrics for automatic evaluation (such as BLEU or TER for machine translation). In order to bridge the gap between those two kinds of approaches to evaluation, a meta-level of evaluation was presented as shared tasks: the challenge is to build an automatic evaluation metric which performs on par with manual metrics–as for example (Machacek et al. 2014). The notion of performance in that domain consists in having a good correlation with human judgments. Most metrics that emerged from those shared tasks have in common that they rely on machine learning (often a simple regression) trained on a set of human assessments from previous years competition on the target task, and generally combine multiple existing metrics, enhanced domain-specific features and the optimization of a loss-proxy to correlation.

2.4.3 Significance testing

Natural language processing systems are typically developed using a finite annotated dataset. The dataset is split in training, development and test (or held-out) shards. While the training subset serves finding model parameters, the development set is used to compare models (and search for good hyper-parameters), and the test set is kept untouched until the evaluation of the system. All three subsets are a sample from the underlying distribution of the data. Each of them is assumed to be representative of the data. However, due to not knowing ways of measuring representativness, a different choice of one of those sets might yield different results, and in particular give a different ranking across systems. In addition, random choices when training ML algorithms might also lead to different evaluation scores: different weight initialization, different training set traversal ordering, number of training rounds, etc. are all known to affect the performance of the resulting system.

Significance testing consists in estimating the probability of being incorrect when we argue that one system is better than another one. We generally confront two hypotheses: $\mathcal{H}_0$, the null hypothesis under which the result comes from a random effect[10], and $\mathcal{H}_1$, the alternative hypothesis under which the result is true. Significance testing computes $P(\text{accept } \mathcal{H}_0)$, and compares it to a significance threshold p_0 defined prior to the analysis. The fact that system A is better than system B is considered statistically significant if $P(\text{accept } \mathcal{H}_0) < p_0$, the probability of accepting $\mathcal{H}_0$ being called the p-value. In the literature $p_0 = 0.05$ (corresponding to a "confidence of 95%" – an entertaining read on the origins of the threshold can be found in (Cowles et al. 1982)) is predominantly chosen, but (Søgaard et al. 2014) argues that we should use $p_0 = 0.0025$ to actually implement a policy where the research community makes mistakes in less than 5% of findings. Significance testing is a topic often debated in science because assumptions associated with significance tests are often not verified in a study[11], and because over-relying on significance testing leads to methodological issues and bias in research policy building (Leek et al. 2017).

Three types of significance tests are described in (Dror et al. 2018) by order of statistical power. Parametric tests (such as t-test) assume a type of distribution for the metric at hand, which is not always compatible with currently used metrics, such as ROUGE. Sampling-based non-parametric tests do not have those assumptions but require a large amount of processing and therefore might not be applicable on larger datasets. Finally, sampling-free non-parametric tests are less computationally intensive but also the least statistically powerful (incorrectly considers the $\mathcal{H}_0$ as true while $\mathcal{H}_1$ is).

In the following, we present two tests: approximate randomization which is a non-parametric sampling test (Riezler et al. 2005), and the modified Wilcoxon signed-rank test which is a non-parametric sample-free test (Oyeka et al. 2012).

[10] The actual wording of the hypotheses depends on the metric and the statistical test used.

[11] Dror et al. (2018) argue that the minimal elements of NLP datasets (sentences, paragraphs, documents...) are not independent because they might have been produced by the same author, reviewed by the same editor, or are related to the same event. Therefore, none of the significance tests available today shall be used.

Approximate Randomization The approximate randomization test is often advocated in the NLP community because it is very easy to implement and does not have the assumption and bias problems of many tests (Noreen 1989; Riezler et al. 2005). This test directly estimates the probability that two system outputs come from the same distribution, irrespective of which system is better than the other. It assumes that the two system outputs can be stratified along a parallel partition (for instance sentences or documents), and that strata are independent. The test consists in sampling from the set of all combinations of strata, and counting the number of times the score difference between the samples is larger than the score difference between the system output. The exact version of this test which evaluates all permutations of the test set is called Pitman's permutation test (Pitman 1937). The p-value can be deducted from the ratio between that number and the size of the sample. Typical sample sizes are in the ranges 1,000–10,000. Algorithm 1 presents an overview of the approximation randomization algorithm for computing the p-value of a difference between two system outputs. A baseline implementation would call the evaluation procedure repeatedly, which is expensive if it includes an alignment process or inference (such as running an NLP tool), but it is possible to cache intermediary results, exchange them at the swap stage, and lazily recompute the aggregate metric (such as F-Score, ROUGE or BLEU). This approach was implemented in SRLEVAL (Favre et al. 2010a†).

Data: A and B, two system outputs on a parallel partition P; $e(\cdot)$ an evaluation function; N the number of samples (typically 10,000); $rnd() \in [0;1[$ a random number generator.
Result: p the estimated p-value.
$r \leftarrow 0$;
baseline difference: $b = |e(A) - e(B)|$;
foreach $n \in [1;N]$ **do**
  foreach $i \in P$ **do**
    if $rnd() > \frac{1}{2}$ **then**
      swap $A[i]$ and $B[i]$;
    end
  end
  difference of sample: $d = |e(A) - e(B)|$;
  if $d \geq b$ **then**
    $r \leftarrow r + 1$;
  end
end
p-value: $p \leftarrow \frac{r+1}{N+1}$;

Algorithm 1: Approximate randomization for significance testing.

This algorithm has two main drawbacks: the nature of the sampling process might result in different significance levels for different initialization, and the assumption of independence between strata is generally wrong with respect to the input (sentences follow each other), but mostly right with respect to systems (sentences are effectively processed independently, or at least strata can be designed to encompass the span of input effectively used by the systems). Another potential problem is that the larger the strata, the higher the p-value. Even if a system is significantly different from another at the sentence level, it might not be so at the document level. Investigating this phenomenon and how to prevent it from happening would be interesting, but might require advice from statisticians.

Modified Wilcoxon signed-rank test The Wilcoxon signed-rank test is a largely used non-parametric, non-sampled significance test (Wilcoxon 1945). While it does not make assumptions on the distribution of the metric used to evaluate systems, it assumes that the metric is continuous and that there are no instances with identical scores (no-difference samples). The modified Wilcoxon signed-rank test addresses these issues and is more suitable for NLP (Oyeka et al. 2012). Like other statistical tests, it assumes that instances are independent.

Let n be the number of instances in the test set (e.g. sentences). Let $b = e(A) - e(B)$ be a vector of differences between evaluation measures for systems A and B on the instances. First, we need to compute $|b|$ the absolute value of the differences between the systems, and $\text{sgn}(b)$ the sign of that difference:

$$\text{sgn}(b_i) = \begin{cases} 1 \text{ if } b_i > 0 \\ -1 \text{ if } b_i < 0 \\ 0 \text{ otherwise} \end{cases} \tag{2.70}$$

Let $r_{|b|}$ be the rank of the absolute values in b[12]. We define $\bar{r}_{|b|}$ as the average rank for each value of $|b|$ – that is if a value occurs 3 times in $|b|$, then the associated average rank is the sum of the ranks divided by 3. Subsequently, p^+ and p^- are the probabilities of differences of being positive or negative.

$$p^+ = P(\text{sgn}(b) = 1) \sim \frac{\text{count}(\text{sgn}(b_i) = 1)}{n} \tag{2.71}$$

$$p^- = P(\text{sgn}(b) = -1) \sim \frac{\text{count}(\text{sgn}(b_i) = -1)}{n} \tag{2.72}$$

Then, the z-score statistic can be computed with the following formulas (see (Oyeka et al. 2012) for details).

$$T = \sum_i \text{sgn}(b_i) \cdot \bar{r}_{|b_i|} \tag{2.73}$$

$$\text{Var}(T) = \frac{n(n+1)(2n+1)}{6} \left(p^+ + p^- - \left(p^+ - p^-\right)^2\right) \tag{2.74}$$

$$z\text{-}score = \frac{T - \frac{n(n+1)}{2}}{\sqrt{Var(T)}} \tag{2.75}$$

From the z-score, one can use a lookup table[13] to retrieve the corresponding p-value and compare it to the significance threshold p_0.

This test is suitable for very large datasets where it is not tractable to apply sample-based tests. It is unsuitable for small sample sizes.

2.5 Conclusion

This chapter was dedicated to an overview of major machine learning models applied to solve natural language processing problems. A broad range of NLP tasks can be cast as one of five problems: classification, regression, segmentation, relation detection and generation. We presented models in which those NLP tasks can be cast: the perceptron algorithm, support vector machines, conditional random field, and various deep learning architectures.

There are several aspects that we did not cover but are important when developing NLP systems. The early parts of the TCGASE loop, corpus creation and annotation, are very important, time consuming and yet important tasks. The problem of annotators diverging in term of interpreting annotation guides or becoming inconsistent in time, the compromise between choosing annotation categories representative of the linguistic phenomenon, or categories easy to predict for a system, the cost of using experts or the problems with leveraging crowd sourcing, are all problems that need to be tackled when building NLP systems. Discussing those issues could easily transform this document into a long book. They are often overlooked by machine learning practitioners who tend to focus on systems in order to maximize performance, and not reconsider the data.

[12] By sorting $|b|$ by ascending value and noting which one is first, which one is second and so on. The ranks start at 1.

[13] With statistics software or online calculators such as https://www.socscistatistics.com/pvalues/normaldistribution.aspx, retrieved on 2019-04-30.

Machine learning is a constantly evolving field and advances in expressiveness and performance of models continues to generate fascinating approaches in NLP. However, one may wonder whether the current trend for ML to conquer most corners of NLP is sustainable and will eventually solve all of NLP.

Chapter 3

Beyond machine learning assumptions

3.1 Introduction

Natural language processing systems are built on models that make various assumptions. The basic assumption of machine learning methods is that training data is representative of test data, and in particular that training data and test data come from the same distribution. This assumption entails that training and test data are representative samples drawn from that distribution, and some machine learning algorithms also assume that instances (words, sentences, documents) are independently and identically distributed (IID). In particular, inference and learning over one instance is performed independently of the content of other instances. This has far-fetching consequences on the capabilities of NLP systems, and requires that researchers maintain a precise understanding of the modeling assumptions linked to the models they use. Taming those assumptions, and overcoming the ensuing limitations has generated a lot of work in the NLP community. Beyond independence, recent models also make assumptions on the dependency structure assumed over the problems, from independence to all sorts of approximations for modeling co-variates.

In this chapter, we go beyond machine learning assumptions to overview different types of dependence assumptions from an NLP point of view, and illustrate them in the framework of specific contributions. As we will see, dependency is often a matter of compromise. In the following, the first assumption explored is the scope of features used to characterize a decision, which are often limited in their nature. The second assumption is that of decision-space independence assumptions linked to the structured nature of natural language, such as the Markov process assumption. The third is the conditional independence of higher level NLP tasks on lower level tasks which is at the heart of the pipeline approach.

3.2 Input representations

It is well admitted in empirical research that an effect, however strong, should not be presumed to only come from the observable captured in the first place. One has to argue thoroughly on the possible sources of an effect, and measure appropriately their incidence on the studied phenomenon.

Machine learning researchers often take for granted that input features readily cover the phenomenon to be modeled and focus on algorithms by relying on performance differentials in order to assess their benefits. It is, however, clear that if input features are lacking in terms of explaining the studied phenomenon, any supervised method for training a predictor will be performance-bound by the set of features used to model the data. This performance limitation problem might actually be due to the following issues: not enough training data was used for training the model (the training set is

a too small sample of the general distribution it was drawn from, and therefore the estimation of the model will be too crude), the model cannot fit the shape of the data for structural reasons (such as a linear model in the context of learning a non-linear function), or the features used as input cannot represent accurately the phenomenon (such as discriminating colored crayons in a black-and-white image). Here, we are interested in that last problem.

A starting point could be that if a human can label an instance, then assuming machine-learning algorithms are human-like, the features extracted from that instance should be enough to reach human performance. A limitation of that assumption well-known to the NLP community is that humans possess an experience of the real world, and some problems can only be tackled with so-called world knowledge, which could be a formal description of all the knowledge amassed by a human over her life. Word-sense disambiguation, textual inference, general dialog systems, or translation from Chinese to a western language are said to be problems that can only be solved with world knowledge. Notwithstanding the issue of formalizing world knowledge, which is an open problem, gathering more training data might be a costly way of addressing world knowledge by allowing the algorithm to capture it indirectly through the data distribution. Another problem of the human-performance hypothesis is that for a range of interesting NLP problems, humans are not able to agree on a labeling (sentiment analysis and emotion detection are classical examples of that issue).

Assuming that an instance is well modeled by a set of features indirectly implies that it is independent from other features one might collect to represent it. There are various cases of input representations that are limited to a set of features while they could encompass a wider context. Understanding spoken content is one of those tasks where research has concentrated on processing the transcript extracted from the audio signal while forgetting that speech occurs in a multimodal context, and that other modalities might be of use.

The modality of choice for natural language processing is text. A large body of human communication comes in the form of textual content, be it books, letters, and newspapers. Even if they are being progressively replaced by electronic counterparts, their basic content is still textual representations of human ideas. Yet, most communication occurs in a multimodal setting where text is not the canonical channel for conveying information. Natural language is by essence multimodal as face-to-face communication benefits for example from non-lexical cues such as gestures, face expressions, social signals like convergence, etc. Assuming that the text modality is independent of the other modalities is naive, and in particular, even if understanding text content is possible for humans without considering other modalities, ambiguities which are hard to solve for systems may be much more accessible when taking into account the whole multimodal context of a message. It is even more interesting when the application is to understand high-level events in a situation.

3.2.1 Multimodal conditioning

The first example we are presenting consists in conditioning a decision with multimodal co-variates which are not apparent to the system designer yet have the potential to bring large performance improvements.

Video understanding involves a typical multimodal scene which comprises images (and potentially texts on images), audio, and transcribed speech as basic modalities. The REPERE challenge organized by DGA, LNE and ELRA consisted in identifying at any time in a TV show all the persons on the screen. This challenge was created to foster multimedia indexing applications such as being able to search for all videos involving a given person, not only in textual mentions but audio and video appearances as well. The task is challenging because in TV shows persons are identified at key moments but their identity is not recalled for long times. For example, a talk show with multiple invited speakers might contain a presentation of the speakers by the anchor at the beginning and then display their name from time to time on screen when they talk. To be able to infer at every instant who is who, systems have to rely on unsynchronized long-distant cues, and rely on propagation to associate an identity with faces in the image modality and speech in the audio modality.

At the time of the REPERE challenge, the standard approach consisted in extracting identities in

each modality (displayed texts, pronounced names, speech segments, faces) based on optical character recognition, speech-to-text, speaker recognition and face recognition. The system created by the PERCOL team[1] can be summarized as follows:

- For displayed texts, zones of interest are processed by an off-the-shelf OCR in all the frames they appear in to create confusion networks of the potential characters. A few rules separate names from other content such as news or stocks, and names are normalized according to a large quantity of crawled names deemed potentially relevant (such as political figures, celebrities…)
- Pronounced names are detected in recognized speech, but this is not robust enough because of frequent errors and out-of-vocabulary names. Therefore a spoken term detection system is leveraged to detect potential names, from the large list, in phonetic sequences. On top of detected names, an attribution system decides whether a detected name is that of the current speaker, next speaker, previous speaker or none of those based on linguistic features from the transcript and structural features from the conversation.
- A large number of speaker models are created with standard techniques such as i-vectors from collected external data and agglomerative clustering in order to perform direct speaker identification.
- The face processing system does not perform face identification[2], but rather detects shots and faces, and clusters them according to their pixel content.

Alone, each of these mono-modal systems are relatively poor: exploiting displayed texts has a high precision but low recall, ASR is noisy and involves an additional difficult step of attributing names to persons who might not be present on the screen or heard, speaker identification works well but the coverage of biometric model dictionaries is not large enough for it to generalize, and the vision pipeline itself is not able to produce identities. Therefore, the system leverages propagation strategies to transfer identities detected in one modality to another modality:

- rule-based cross-modal propagation strategies which use the result of clustering in a given modality to propagate names to another modality;
- a multimodal clustering technique which refines the clustering in each modality to be able to transfer labels across modalities (Bendris et al. 2014†);
- a final fusion step which merges the results from other strategies.

At the REPERE competition, given the difficulty of correctly attributing identities in each of the modality, this approach was not able to outperform the systems created by the other participants. Yet, for the last iteration of the competition, we decided to leverage a bias of the dataset which is that most TV shows occur both in training and test, and therefore often involve the same anchors, are shot at same or similar sets, and have a very regular structure. We used this strategy to identify the background of camera shots and rely for instance on the fact that some cameras are subshots of others in order to create show-specific propagation rules which would allow for instance identifying the invited speaker on the left of the table and label her with a displayed name from the beginning of the show. This kind of strategy, and an aggressive model selection based on performance analysis on the development set and a leave-one-out of the training set lead to the PERCOL consortium winning the competition. Even though the proposed approach does not generalize well to new TV shows, it exemplifies the fact that in a multimodal context, a system should look at the whole picture and try to understand the context in which the instances have been generated.

Even though the "understanding" problem addressed by the aforementioned work is a difficult one, the language part is relatively underdeveloped because linguistic signals for identifying people in TV shows are weaker and harder to address than other signals. Yet, there are other applications where this link is tighter and can be explored.

[1] In which I was involved.
[2] The team did not have reliable enough technology for that task.

3.2.2 Multimodal embedding fusion

The second example seeks the creation of multimodal representations from several monomodal inputs, in order to inform the decisions of a system.

The multimodal understanding task at stake is the detection of the role of speakers in a TV show (Rouvier et al. 2015a†). For each speaker turn, the system has to predict whether the speaker is the host of the show, a commentator, a reporter or another role such as the interviewee. This task has been typically solved only looking at one modality at a time. The presented work extends monomodal systems with the idea of multimodal embedding fusion, which is an alternative to early (input-level) or late (decision-level) fusion. The text, audio and image modalities are processed[3] as follows:

- Text: a convolutional neural network detects word-embedding n-grams similar to typical features extracted in non-neural systems (Dufour et al. 2011). It assumes that professional speakers use discriminative word patterns typical of their role (such as asking questions for the host, or using a more colloquial style with disfluencies for interviewees).

- Audio: an i-vector speaker verification system generates a fixed-size representation of the audio (which should contain the speaker identity, but which actually reflects a lot of factors of the input, such as the acoustic environment). This representation is then fed to a fully connected MLP. This system is supposed to model idiosyncratic effects of the behavior of speakers, as well as the environment in which speech was captured (for instance, interviews often take place in the field while hosts and commentators are on a set).

- Image: a standard Alexnet CNN produces a representation of one image per segment (the median), being able to model the outfit of the speaker, the number of persons on the screen, the presence of overlaid text, and the overall objects depicted. This approach (not detecting an actual person) is also biased towards modeling the environment because the speaker might not be seen on the screen, and it is hard to link an image of a speaking person to actual speech such as with lip sync for instance (Chung et al. 2016).

All three systems are pre-trained with standard tasks such as ImageNet for the image part (Krizhevsky et al. 2012), word embeddings (Mikolov et al. 2013) for the text part, and speaker identification among 1,000 speakers for the audio part (Rouvier et al. 2015b†). Each of them is then trained on the role classification task separately. Two multimodal fusion strategies are compared: late fusion which ensembles the decision scores of the systems (sum of output), and a "multimodal embedding fusion" which uses the penultimate layer of each system as rich representation for the corresponding modality and is then fed to a multilayer perceptron to combine them in a final decision. Results show that the multimodal systems improve over the monomodal systems, and that embedding fusion, by having access to multimodal regularities, leads to an additional improvement (Rouvier et al. 2015a†).

An additional fine-grained example of leveraging multimodality in the context of language understanding is the task of resolving prepositional phrase attachments in an image caption with features coming from an analysis of the image itself (Delecraz et al. 2017†). PP-attachment is notoriously hard syntactic paring task because selecting the word which governs a prepositional phrase depends on semantic interpretation of the sentence. For example, the sentences "I am holding a cup with a blue glove" and "I am holding a cup with a blue bottom" have both a very similar structure, and in particular the same sequence of part-of-speech tags, yet in the first one, the PP-phrase introduce by "with" is attached to the verb "hold" as the glove is an instrument for the action to hold, and in the second one, the PP is attached to the noun "cup" as the bottom is a part of the cup. There are other examples which cannot be disambiguated only from world knowledge such as "I am looking at the man with a telescope" where both interpretations of who is operating the telescope are valid without further context. In the world knowledge case, parsers can learn some of the associations in their training corpus, but they do not generalize well and efforts to leverage lexical resources or regularities from large corpora lead to small gains (Agirre et al. 2008; Mirroshandel et al. 2012).

[3]Each audio segment is the base for an instance in which features are extracted.

3.2.3 Alignment across modalities

The last example focuses on making correspondences between phenomena across modalities, bridging elements of each modality together while leaving the rest unrelated. It can be seen as an alignment process across modalities.

When an image depicting the same scene as described in a sentence is available, a human can readily resolve the ambiguity, so a system should be able to benefit from this additional context. The Flickr30k-Entities corpus contains about 30,000 images with multiple captions crowdsourced originally to build a paraphrase corpus, and later extended with hand-annotated entities with corresponding bounding boxes in the image, semantic categories (such as human, animal, instrument…), and coreference links across modalities and captions of a same image. We additionally annotated gold-standard PP attachments on 29k potentially ambiguous prepositions. A dependency parser trained on a conventional tree bank generates attachments with 75% accuracy for that corpus and training a reranker given lexical features can raise that accuracy to 86% (due to the mismatch between the domain of the Penn Treebank and image captions). Such a reranker would not generalize well on a corpus involving objects with different properties, such as from the medical domain, and therefore we explored the possibility to generate visual features and use them in the reranker in place of lexical features. The approach consists in detecting objects in the image, linking them to textual entities in the sentence, and compare their relative position and semantic categories:

- Visual object bounding box detection and semantic class prediction is performed with the YOLOv2 (Redmon et al. 2016) retrained on the Flickr30k objects.
- Textual entities are detected with a simple BIO tagger based on an averaged perceptron.
- A matching score between phrases and detected objects is computed by projecting both in a common representation space through neural networks, ResNet-152 for the content of object bounding boxes, and GRU recurrent networks for phrases (Faghri et al. 2017).

Then, from this pipeline, features based on the relative position of the objects corresponding to the phrases being mediated by a preposition are leveraged to train the reranker who decides whether a potential PP attachment is correct or incorrect (in the case of attachments to verbs, the link is forwarded to the subject/object on the other size of the verb). Results show that visual features can improve PP-attachments by 3 points in average, but by a very large margin for locative prepositions such as “near”. Even though lexical features offer a better improvement and the conjunction of lexical and visual features is not better than lexical features alone, this result represents a landmark because detecting objects in the visual modality is hard (recall of 50%) and matching them to phrases is also hard (20% error rate on gold-standard bounding boxes).

To conclude this section on representation-level independence, considering that text is independent of other modalities is inaccurate and an important body of work has shown that cross-modal information is relevant. A recent example is the shared task on multimodal machine translation (Specia et al. 2016). Nevertheless, the challenges of multimodal processing are numerous, in particular the varying performance of predictions in different modalities, the difficulty to build end-to-end multimodal systems, and the lack of synchronization between phenomena which generates long-range dependencies difficult to model.

3.3 Structured Predictions

NLP problems often involve predicting multiple labels per instance, such as associating one part-of-speech label per word in a sentence. Under a holistic model, the POS label for a given word depends on *understanding* the whole sentence, which means that all words from the instance should be used for each prediction. However, in practice, often a subset of the input is considered, and there is a potential benefit in looking at predictions our system would make for neighboring slots when predicting a given

label. Structured predictions are prediction problems where some form of dependency between the predicted labels is assumed. This assumption might be useful because of some modeling compromise (such as only looking at the current word), or due to some constraints on the validity of produced labeling (in parsing, the output must be a tree and shall not be a disconnected graph; in named entity tagging with the BIO encoding, an "inside" label cannot follow an "outside" label).

In this section, we explore the compromise between rich modeling and dependency structures in the output, as well as the problem of exact versus approximate inference when the dependency structure leads to difficult inference problem solving.

3.3.1 The Markovian assumption

Markov processes are stochastic processes in which the conditional probability distribution of future states only depends on the current state. They have been popular in NLP for two reasons: they restrict the space of events for which we need an accurate probability estimation to states and state transitions, and they allow to decompose structured prediction problems and therefore to infer exactly maximal probability assignments thanks to dynamic programming. While Markov processes work on chains of events, they can be extended to trees or graphs (Markov random fields), and the notion of current state can be extended to a clique in the graph with a bounded size (order-n Markov structures).

The Markov independence assumption has been the "cash cow" of NLP for a long time because the benefit of exact inference of structural predictions was higher than that of modeling long-range dependencies, probably due to rather poor modeling of emission and transition probabilities. Speech recognition is famous for having been dominated by HMMs for phonetic predictions, and n-gram language models for word sequence predictions. In ASR, the fate of HMMs was sealed with the proposal of first end-to-end approaches (Hannun et al. 2014; Collobert et al. 2016) which learn to generate a sequence of characters directly from speech frames or audio samples. While best performance on benchmarks are still occupied by HMM systems, and even end-to-end systems benefit from language model rescoring (Collobert et al. 2016), the gap is small, and it seems that engineering progress is the only barrier to the retirement of HMMs in those applications. The machine translation community has followed similar trends while going from phrase-based systems to Neural Machine Translation (NMT). In both speech and translation applications, inference was approximate because of the size of the state space, and even with current neural-network-based systems, beam search is still beneficial although conditional independence is not true anymore (Klein et al. 2017).

Part-of-speech tagging and syntactic parsing have followed a similar trend which is to put less emphasis on the decoding process and more on the modeling part. For POS tagging, the trend is straightforward by going from HMMs (Kupiec 1992) to log-linear models (Collins 2002) to CRFs (Awasthi et al. 2006) and finally to sequential neural networks such as LSTMs (Ma et al. 2016). In parsing, the trend is less straightforward because the variation in popularity of formalisms (constituents, dependencies, adjoint tree grammars…) and decoding strategies (chart parsing, graph-based or transition-based parsing). Yet the community has followed a similar trend with probabilistic context-free grammars (PCFGs) which are HMMs on context-free grammars, refined as PCFG-LAs which include latent variables for refining non-terminals (Petrov et al. 2007), and a series of models supported by the averaged perceptron (McDonald et al. 2005b; Bohnet 2010) and recently by neural networks (Dyer et al. 2015). As an example, the system that Stanford submitted to the CoNLL'17 parsing shared task consisted of LSTMs which create representations at word indexes then combined to create dependency-level representations fed to decision layers. This approach effectively creates contextual representations based on the whole sentence, so that syntactic phenomena can readily be predicted without accounting for other decisions: the authors of the system describe a rule-based system for ensuring that predictions lie on a tree, and postpone using a maximum spanning tree decoder as future work.

Independence assumptions, in a probabilistic sense, allow for exact inference of the structure probability maximization problem. Removing them with end-to-end and recurrent neural networks helps growing a much finer modeling of singular events themselves, and therefore be less reliant on capturing the global structure. Yet, even if in a probabilistic interpretation neural networks can avoid many

conditional independence assumptions, they make other kinds of assumptions similar to independence based on their topology, and the kinds of non-linear operations they perform, for which we don't have yet the mathematical tools which will help us understand how they affect system performance.

3.3.2 Exact inference

Predictions in NLP often involve an inference step which searches for the structured output with maximum score. In tasks such as part-of-speech tagging or syntactic parsing, one may leverage dynamic programming through variants of the Viterbi algorithm to find the exact solution to the inference problem. When the search space becomes large, such as in automatic speech recognition or machine translation, the DP search space is pruned in order to explore only the most promising solutions. There are other tasks, such as summarization, for which inference does not reduce to a dynamic program and for which exact inference is harder.

Automatic summarization is often cast as a constrained optimization problem: pack as much information as possible in a limited size summary. In extractive summarization, this problem leads to models which select subsets of the input, such as sentences, until the size limit is met. If each sentence brings a value to the summary but adds its length towards the length constraint, this problem is a knapsack problem, a well-known NP-complete problem (Karp 1972). For this reason, a lot of summarization implementations turn to approximate inference, such as greedy search for the solution.

For instance, the seminal work on maximal marginal relevance (MMR) proposed to select a set of sentences which maximizes total relevance while minimizing redundancy (Carbonell et al. 1998). Relevance of a sentence is modeled as its cosine similarity with the centroid of the input document to be summarized (the average sentence vector), and redundancy as the cosine similarity between two sentences in the summary. Ideally, the summary should contain sentences which maximize their individual relevance score while minimizing the sum of their pairwise redundancy scores. This problem is akin to a quadratic knapsack problem (Gallo et al. 1980) which is expensive to solve for large inputs, and therefore the MMR algorithm yields a greedy solution which consists in iteratively selecting the sentence which maximizes the following score:

$$\hat{s} = \underset{s \notin S}{\operatorname{argmax}} \left(\lambda cosine(s, c) - (1 - \lambda) \max_{s' \in S} cosine(s, s') \right) \tag{3.1}$$

where S is the set of sentences selected so far in the summary, c is the centroid of all input sentences, cosine$(\cdot, \cdot)$ is a similarity operator in the sentence representation space, and λ is a hyperparameter balancing the contribution of relevance and redundancy[4]. This greedy search assumes that selecting a sentence at a given iteration is independent from the sentences which will be selected in the remaining iterations. Like MMR, a lot of extractive summarization approaches rely on a fancy estimation of sentence relevance – LexRank (Erkan et al. 2004), ML-based approaches (Kupiec et al. 1995), SVD (Gong et al. 2001) – yet compute redundancy as word overlap with the already selected sentences. Sentences that do not increase the lexical diversity of the summary at a given iteration of greedy selection are discarded.

The greedy summarization approximation is not such a bad approximation, as evidenced by the fact that summarization objectives can be submodular functions over sets, which have been shown to be maximizable by a greedy algorithm with a guaranteed approximation factor (Lin et al. 2011). A function f is called submodular if for any $A \subseteq B \subseteq S$ and $s \in S \setminus B$:

$$f(B \cup \{s\}) - f(B) \leq f(A \cup \{s\}) - f(A) \tag{3.2}$$

which means that the value gained from adding a sentence s to a set of selected sentences does not increase as the set of selected sentences grows. A relaxed version of MMR (which models redundancy as the sum of similarity of the selected sentence to all other sentences) can be written as a submodular

[4]Typical values of λ for news multidocument summarization are around 0.95

function, and in particular, if a summary valuation function is monotone ($f(S \cup \{s\}) \leq f(S)$), the following greedy algorithm can be applied:

$$\hat{s} = \underset{s \notin S}{\operatorname{argmax}} \frac{f(S \cup \{s\}) - f(S)}{l_s^\lambda} \tag{3.3}$$

where S is the summary so far, l_s is the length of sentence s and λ is a scaling factor [5]. This essentially selects the sentence with highest objective gain to length increase ratio. In practice, (Lin et al. 2011) show that greedy search for a selection of sentences with this method results in close to optimum objective maximization, and increased ROUGE score compared to baselines.

Yet, another trend of work looks at exact solutions to the sentence selection problem. (McDonald 2007) proposed to formalize the same MMR variant (sum instead of max in redundancy computation) as an Integer Linear Program (ILP) which can be maximized with off-the-shelf solvers. They proposed the following formulation:

$$\max \sum_i x_i \text{Rel}_i - \sum_{i<j} x_{ij} \text{Red}_{ij} \tag{3.4}$$

$$s.t. \sum_i x_i l_i \leq L \tag{3.5}$$

$$x_{ij} \leq x_i \quad \forall i, j \tag{3.6}$$

$$x_{ij} \leq x_j \quad \forall i, j \tag{3.7}$$

$$x_i + x_j - x_{ij} \leq 1 \quad \forall i, j \tag{3.8}$$

$$x_i \in \{0, 1\} \quad \forall i \tag{3.9}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \tag{3.10}$$

where Rel_i is the relevance of a sentence i, Red_{ij} is the redundancy induced by having sentences i and j in the summary, x_i is an indicator that sentence i is in the summary, x_{ij} is an indicator that sentences i and j are together in the summary, l_i is the length of sentence i and L is the length budget for the summary. The other constraints ensure the consistency between x_i and x_{ij} variables. This linear formulation for a quadratic problem induces a quadratic number of variables and constraints which quickly lead to large instances.

Another global formulation is presented in (Gillick et al. 2009a†). It relies on the idea that sentence-level computation of relevance and redundancy might not be appropriate since some sentences contain less relevant clauses, and information in one sentence might be redundant with another sentence while both cover valuable content. The summarization problem is then cast as a budgeted set-cover problem where sentences cover information units (here, word bigrams), and the budget operates on sentence lengths.

$$\max \sum_i c_i \text{Rel}_i \tag{3.11}$$

$$s.t. \sum_j x_j l_j \leq L \tag{3.12}$$

$$x_j \leq c_i \quad \forall j \lhd i \tag{3.13}$$

$$\sum_{j \lhd i} x_j \leq c_i \quad \forall i \tag{3.14}$$

$$x_i \in \{0, 1\} \quad \forall i \tag{3.15}$$

$$c_j \in \{0, 1\} \quad \forall j \tag{3.16}$$

[5] $\lambda = 1$ leads to an approximation factor of $1 - e^{-\frac{1}{2}}$, but other values of λ can lead to higher ROUGE scores.

where Rel_i is the relevance of information unit i, l_j is the length of the sentence j, x_j is an indicator that sentence j is in the summary, c_i is an indicator that information unit i is in the summary, and $j \triangleleft i$ indicates that sentence j covers information unit i. This model was built while trying to generate summaries maximizing ROUGE-2 score, the official metric of the DUC/TAC evaluation campaign. It is an approximation[6] of ROUGE-2 where the relevance score Rel_i is the number of reference summaries a bigram occurs in. For multidocument summarization, replacing Rel_i with the number of documents a bigram occurs in lead to one of the best systems at the TAC 2008 and 2009 evaluation campaigns[7].

While ILP is an attractive formalism for inference in NLP, it has to be noted that even very strong implementations can be slow on hard or large instances. For example, formalizing the Viterbi algorithm with ILP leads to a very slow decoder while the classical dynamic programming approach is much more efficient. The budgeted set-cover maximization formulation is submodular monotone, given non-negative Rel_i scores, and therefore also benefits from good greedy approximation (Khuller et al. 1999). Local search is another relatively general approximation technique which randomly swaps sentences in the solution in order to improve the score of the solution. It that can solve large instances of the budget max-cover and modified MMR formulations almost always to optimality in a short time[8].

Inference with non-polynomial formulations also brings challenges in the context when composing hypothesis spaces. How can the (exponential) space of possible solutions of a summarizer be represented? How can n-best lists be extracted efficiently? Those are open problems which might turn very hard to solve.

To wrap up, assuming independence between objects involved in the inference process of an NLP task may lead to reduced performance compared to exact inference while providing faster inference (in the greedy case). This is especially true for summarization which is by nature a problem where constraints induce NP complexity, even though quality greedy approximations are possible for a restricted set of problem formulations.

3.4 Independence at the Phenomenon Level

This section outlines the type of assumptions made at the phenomenon-level, by assuming for example that syntax does not dependent on semantics. It is embodied in processing chains which produce several levels of annotations which sequentially dependent on the previous levels. We look into how processing chains are implemented, the kind of limitations they impose on the predicted phenomena, and a few extensions that have been proposed to overcome those limitations.

3.4.1 Processing Chain

The historical approach to natural language processing consists in modularizing the prediction of various levels of analysis as a processing chain which sequentially extracts higher levels of phenomenon representations from the input from lower-level representations. For example, to perform surface semantic parsing, such as semantic role labeling, one will typically run the following steps:

- sentence splitting (from raw text to sentences);
- word tokenization (from sentences to tokens);
- part-of-speech tagging (decorate tokens with part-of-speech tags);
- lemmatization (decorate tagged tokens with the lemma they correspond to);
- syntactic parsing (predict the syntactic structure of a sentence given tokens, part-of-speech tags and lemmas);

[6] It is an approximation because ROUGE clamps occurrences and the metric is resampled with jack-knifing.
[7] Part of the success also came from high quality sentence splitting, and post-processing rules (Gillick et al. 2009b†).
[8] Unpublished personal result implemented with `localsolver` (Benoist et al. 2011)

- and finally semantic parsing (predict a semantic interpretation of the sentence given syntactic structure and the other features of tokens).

Additional tasks are often considered, such as named entity tagging and linking – marking token sequences with direct references to real world entities such as persons, organization, places, time references or amounts (Galibert et al. 2011), and resolving their identity by linking them to entity databases such as freebase (Bollacker et al. 2008) –, coreference resolution – linking together textual references to entities such as pronouns, nominal phrases or named entities), or word sense disambiguation (finding for each word in a text its actual sense given a sense inventory. In a more general framework, NLP tasks often require as input elements predicted by other tasks, which has led to the development of processing chains.

Dividing to conquer the NLP processing chain was a wise choice because it allowed to study the problems separately, under a linguistic model umbrella, and come up with well-engineered solutions of the individual problems. The main advantage of modularizing the processing pipeline is that expertise can focus on subproblems, and therefore better leverage linguistics as a source of inspiration for system design, in particular to better control the generalization of each module. For example, a part-of-speech tagging model can benefit from specialized lexicons which list possible part-of-speech labels for each word. Another advantage is that models can have different nature, i.e. rule-based or statistical, depending on the ambiguity of the processed task. They can also be trained on different labeled datasets according to availability, or datasets of different sizes according to the ambiguity of the problem at stake. Finally, modules later in the pipeline can benefit from the output of modules that came earlier in the pipeline and use the generated annotations as their input. Syntactic parsing can benefit from knowing word classes (such as part-of-speech tags) instead of fully modeling lexical relations which would require very large amounts of training data, just to be able to cover a reasonable subset of the lexicon.

Unfortunately, the main problem with this approach is that, given realistic input data, submodules inevitably make mistakes, errors which propagate along the processing chain, and eventually undermine the final result. In particular, a mistake in part-of-speech tagging directly impacts a syntactic parser which relies blindly on the nature of the underlying annotations[9]. For example, in English most verb tokens can be nouns, and an unusual sentence might confuse the part-of-speech tagger to label a verb as a noun, which will preclude the parser from associating it with a subject or an object, and therefore render the syntactic analysis incorrect. In the same way, an error in lemma or syntactic structure will impact semantic parsing, and even an error in sentence splitting (a task which is considered easy but often overlooks the ambiguity of punctuation symbols) can lead to degraded results at the end of the chain. The problem of error propagation is also exacerbated by the so-called snowball effect, well known in the automatic speech recognition community, in which a small error in the early stages of processing also impacts neighboring predictions due to the fact that structured predictions are the result of the combination of features that span the neighborhood of the symbol to be predicted. In automatic speech recognition, this effect is seen when words absent from the lexicon have to be recognized, leading to their misrecognition and a misestimation of the probability of all the n-grams that overlap with them (under an n-gram language model). This generally leads to misrecognition of neighboring words as well. In syntactic parsing, the same effect operates when a word is mistagged as it impacts all the features related to its governor and dependents in first order models, and even further in higher order models.

Another problem with the processing chain approach is that a conditional independence is assumed on subsequent tasks. Sentence splitting is assumed conditionally independent of syntactic parsing, but a syntactic parser could help disambiguate abbreviations that are also sentence endings by comparing the likelihood of the two sentences resulting from the split with the unsplitted sentence (Favre et al. 2008a[†]). Another problem is prepositional phrase (PP) attachment in syntactic parsing which is itself a semantic problem, and could benefit from semantic constraints. Additionally, in spoken sentence segmentation, where a split can occur after each word, attaching a PP to the current sentence or the

[9]The parser could be trained on predicted part-of-speech tags to limit the problem.

next sentence also requires syntactic, semantic and even discursive information, although non-lexical cues such as prosody can help disambiguate the PP attachment when the speaker uses them.

Processing chains following the pipeline approach propagate errors, and preclude the use of subsequent high-level decisions to inform earlier low-level decisions. Fortunately, several techniques can improve upon this assessment.

3.4.2 Composing Hypothesis Spaces

One way to get around the problems of the pipeline approach is to generate multiple decision hypotheses for a given task and pass them as input of the subsequent tasks, postponing the actual decision to downstream processes. This allows to reconsider early decisions when higher-level context is available.

Naive hypothesis spaces can be created through n-best lists (Stolcke et al. 1997). They have been used a lot at the confluence of ASR systems and natural language understanding (NLU) applications. A NLU system is run on each of the ASR n-best hypotheses and an application module decides which hypothesis is best. The problem with n-best lists is that they are very inefficient in a pipeline context as the number of hypotheses grows exponentially with the depth of the pipeline. They also lack diversity, especially in the context of structured predictions or natural language generation where the number of hypotheses grows with the size of the possible substructures or the lexicon, and even large n-best lists may only be variants of very few decisions (Gimpel et al. 2013).

In sequence prediction problems, hypothesis lattices (graphs with a DAG structure) have been proposed as compact representations of decision spaces in the well-studied framework of weighted finite state automata. In particular, the hypothesis space across tasks can be represented as weighted finite state transducers and the pipeline as the composition of a number of transducers, each achieving one task. The MACAON NLP processing chain (Nasr et al. 2011[†]), developed in the TALEP group, is built on this framework and can perform lexical analysis, part-of-speech tagging, named entity recognition and chunking of word lattices. The syntactic and semantic parsing modules it contains cannot, however, process lattices as they generate non-sequential structures and therefore their hypothesis space cannot be easily represented by weighted finite state transducers (WFST), but would rather have to resort to tree and graph automata (Fülöp et al. 2009). Note that decision lattices benefit from the range of pruning and approximation algorithms available for WFSTs and composition can be implemented lazily which leads to efficient space representations and computations in the processing chain (Allauzen et al. 2007).

Another representation for hypothesis spaces was proposed in the ASR community. Confusion networks (CN) are an approximation of lattices which are created by aligning hypotheses to a pivot hypothesis while imposing input constraints (such as time for ASR) and rescore decisions to reflect ambiguity (Hakkani-Tür et al. 2006). This leads to an input-synchronous sausage-like structure where each bucket represents competing decisions, and are only useful when the predicted labels are not synchronous by nature. One benefit of CNs is that they can represent rich hypothesis spaces compactly by adding paths which do not exist in the original decision space, and are therefore less prone to the snowball effect. They have proved effective for NLU applications on top of ASR hypotheses (Tur et al. 2002). Their drawback is that they might generate exponentially many paths when composed in a processing chain because of the exit fanout of the states which is much larger than in a lattice.

The main issue of composing hypothesis spaces is that the system needs a joint model over all the tasks to decide the best path, a decision often left to the end-user application. A kind of naive Bayes approach would consist in considering the tasks as independent and compute the joint model score as the product of the probabilities returned by each task, but this approach is not satisfactory for the reasons aforementioned. Another approach would be to consider the joint model trained explicitly on all tasks, but it would require large datasets annotated in all modalities. In addition, hypothesis spaces for individual modules are built with independently trained models, which might be problematic if the relevant information for deciding which hypotheses to keep depends on decisions that will be made by other modules.

3.4.3 End-to-end models

Recent development in deep learning has given traction to the idea that predicting actual symbolic representations for intermediate tasks (predicting intermediate linguistic representations) might not be necessary, entailing that high level tasks such as machine translation, summarization, speech recognition, information retrieval or dialogue systems, could be learned directly from raw input features. Given enough training data, deep neural networks should be able to learn latent intermediate representations and leverage them to make sensible decisions in the target task domain.

The end-to-end training approach has not been possible prior to the advent of deep learning because non-deep models learn disjoint weights for similar phenomena. Feature-based models such as linear SVMs, Naive Bayes or MaxEnt (or Struct-SVM, HMMs and CRFs for structured predictions) learn independent weights for each feature, and therefore cannot recognize nor leverage features which have a similar behavior because of a shared linguistic phenomenon. Kernel-based methods with non-linear kernels are able to account for the similarity between instances, but not at the feature level. These limitations result in large amounts of annotated data being required to achieve good performance on high-level tasks directly from input features. Models with latent variables, such as LDA (Blei et al. 2003) or Hidden-CRFs (Quattoni et al. 2007), correspond to a first step towards models which learn intermediary representations to model the complex phenomena of NLP.

Each layer in a deep neural network performs a linear transformation of the input followed by a non-linear activation function. Stacking such layers builds non-linear latent representations of the input, thanks to chain-rule differentiation which allows to back-propagate the supervision signal across layers. This results in systems that can learn to build representations and combine them to more abstract representations along layers. The output at each layer can be seen as latent variables which are learned along the system. This was apparent in the computer vision community where the composition of hand-designed features was replaced by deep convolutional networks for single-input single-output end-to-end systems (Krizhevsky et al. 2014). In the NLP community, the variety of structure outputs made the emergence of end-to-end methods less straightforward, but encoder-decoders, especially with attention mechanisms, have been key to building successful end-to-end systems.

For machine translation, in the phrase-based era of statistical MT, word-alignment was considered a latent variable and learned separately of the translation model generally with unsupervised models such as the IBM family of models, in order to devise an independent phrase translation model and use a decoder to find the most likely sequence of phrases to translate a given source sentence (Koehn et al. 2007). The SMT community has put a lot of effort to improve translation models with linguistic features such as syntactic and semantic parsing (Quirk et al. 2005; Wu et al. 2009) and such systems fared well in competitions (Callison-Burch et al. 2011). Neural machine translation completely replaces the different stages of SMT by a single sequence-to-sequence model which reads words or characters one by one from the source sentence, and produces words or characters one by one in the output (Bahdanau et al. 2014). Attention mechanisms over LSTMs or without recurrent units (but multiply stacked) (Vaswani et al. 2017) can learn a latent alignment and often produce better translations than SMT systems.

Automatic summarization has been dominated by extractive approaches which select sentences from an input, while efforts to build abstractive systems which construct novel text resulted in subpart performance. Advances in neural machine translation led to the emergence of neural summarizers that treat the input document as a source language sentence, and generate text conditioned on that input following the encoder-decoder framework, learning to summarize directly from pairs of texts and corresponding summaries (Rush et al. 2015). Early attempts were limited by the length of the input they could consider, and would not generalize to never-seen words. Current attempts at neural summarization include copy mechanisms and estimation of coverage through attention mechanisms (Zeng et al. 2016; See et al. 2017; Nallapati et al. 2016).

Another application where end-to-end modeling is a success in automatic speech recognition (ASR). The traditional ASR pipeline includes parameter extraction from the waveform (typically Mel frequency cepstral coefficients, MFCCs), followed by an HMM modeling of phoneme sequences such as

triphones, a lexicon which converts phoneme sequences to words, and a language model which scores word sequences. Neural networks were used for estimating the HMM emission probabilities (Waibel et al. 1990; Hermansky et al. 2000) and for language model estimation (Bengio et al. 2003; Schwenk et al. 2002), but still in a HMM framework. Only recently have end-to-end methods emerged for speech recognition which learn directly to map parameter vectors to a sequence of characters without explicit supervision on the intermediary representations (Hannun et al. 2014; Amodei et al. 2016; Bahdanau et al. 2016; Collobert et al. 2016).

There are many other examples of end-to-end processing beyond the classical NLP applications presented here and in other fields. The main idea is that feature engineering, which requires expertise and is idealised by linguistic theories that might not completely explain the data, can be learned directly by end-to-end models. In a sense, this approach is a great success, but deep models tend to be black boxes, and the expressive power gained by more model complexity results in a lack of explainability: by not having access to a symbolic representation of intermediary decisions, it is hard to understand and explain system errors. More importantly, the independence assumptions between features assumed by shallow models is still present in deep models in more lenient form. For example, character-level CNNs cannot represent exactly words longer than the size of their input window, yet they are successful in many tasks. Sometimes, constraining model capacity also favors learning general representations, at the cost of less representative modeling assumptions. What kind of linguistic phenomena cannot be represented by a given neural architecture remains understudied and shall be explored more thoroughly by the community. Better mapping neural network architectures with observed linguistic properties might also be an interesting research avenue.

3.5 Conclusion

There are a number of questions left open in the domain of dependency assumptions. For example, we have seen that using the right input representations is very important in NLP and that feature engineering is important to tackle all the aspects of a problem. Even if deep learning will find good feature combinations and drop irrelevant inputs, it is unable to leverage features not available in the input even if they might be easy to capture. This advocates more research in the multimodal setting, to extend typical work on textual representations of language. One crucial question is whether we need to reconsider the value of the human-level sensing assumption, stating that if a human can solve a problem from a set of inputs, the machine should be able to do as well. Our simulations of the human input and cognitive systems are still rather poor, and therefore a system might be blind of phenomena actually perceived by humans. Conversely, some features might look irrelevant to humans while having statistical properties that can be accounted for by ML systems.

In the domain of structured predictions, there seem to exist a strong relationship between modeling power and decoding strategies. An ideal input model would not need any decoding because it would have all elements available to make the right decision on a given item of the output structure, and an ideal decoding process would require only minimal modeling (like log-linear feature models). Yet, the current research seems to choose different compromises depending on which input model or decoding process becomes favorable in the community. The idea of learning this compromise is not new (Daumé et al. 2009; Kim et al. 2017), but it seems evident that future models will embed the sequence of decisions made by the decoding process in the modeling stage, and make a choice of strategy based on the data itself. Policy estimation with deep models in reinforcement learning might be a way of exploring this avenue. Another interesting question regarding the compromise between modeling and decoding is that of the existence of similar compromises in the human brain. Do we sacrifice modeling power on the input for better accounting for distributional properties on the output?

The pipeline approach was very popular in NLP because it allowed to decompose problems to easier subproblems, and to train different parts on different datasets. They were, however, riddled with error-propagation problems and an inability to model jointly all phenomena. Deep learning has brought end-to-end approaches which don't assume the existence of intermediary symbolic representations,

and can build continuous intermediary representations without direct supervision and produce directly predictions for an end task. Yet, end-to-end methods typically require large amounts of training data, even if they can benefit from some sort of pretraining. How to build end-to-end systems that benefit from the same modularization advantages as pipelines? Can we build a good enough understanding of data distributions which would allow us to train systems piecewise on subsets of tasks?

Modeling assumption, and in particular independence, are very important to the success of natural language processing systems that rely on machine learning. We have seen in this chapter that in addition to the classic assumptions made by ML approaches (such as independence between instances), NLP might suffer from forward independence of processing levels, conditional independence in the decision-making process essentially to build efficient inference algorithms, and independence at the representation level, by not accounting for the full context in which a linguistic event occurs. There are solutions to overcome those independence assumptions, such as end-to-end models, global exact inference through ILP, or multimodal processing. Deep learning seems to remove a lot of those assumptions with models such as bidirectional LSTMs which can move information around from potentially long distances. Yet, even if independence is not assumed anymore, the type of transformation performed by the model between intermediary representations is the new assumption. For example, in the computer vision community, CNNs have been proposed to neutralize the position of detected patterns, multilayer and multi-resolution CNNs also neutralize the size of patterns, but there is no practical solution for neutralizing the orientation in 3D space of objects within a 2D image. This kind of assumption is currently addressed by training models with very large quantities of data which span the distribution of object orientations. The question is whether NLP is affected by similar phenomena and if there are invariants that can be correctly modeled in order to reduce the quantity of data necessary to properly cover a phenomenon.

PhD student work relevant to this chapter

- **Olivier Michalon** worked on statistical models for predicting semantic frames. Frame structures model surface semantics as graph structures anchored to text. Generating such structures is a good example of a compromise between modeling and decoding. Olivier proposed various statistical models from log-linear local models which build the structure from bottom to top, to constraint-based global inference through integer linear programming on maximum-likelihood weights. He also proposed representations based on paths in the inference graph to reduce computational requirements while keeping interesting modeling properties (Michalon et al. 2016).

- **Jérémie Tafforeau** worked on multitask inference for natural language processing. His contribution was centered on exploring how deep learning models could be used to reconsider the pipeline approach. He built an end-to-end model based on stacked LSTMs and trained with a multitask loss to perform all tasks of the NLP pipeline from part-of-speech tagging to simplified semantic frame parsing. Jérémie experimented with loss functions in order to build representation spaces for a target task, and explored error propagation in comparison to the pipeline approach, especially in regard to automatic speech recognition (Tafforeau et al. 2016).

- **Sébastien Delecraz** worked on multimodal language understanding. He proposed two contributions relevant to this chapter. The first one is multimodal embeddings, which allow pretraining of deep systems on separate modalities, and merging their hypothesis space at the representation level instead of the typical early/late fusion dichotomy (Rouvier et al. 2015a). The second one looks specifically at naturally ambiguous syntactic decisions which cannot be resolved only using text and have to resort to external knowledge. He proposed a method for generating image-level representations and leveraging them in the context of a syntactic parser for improving prepositional attachments that can only be resolved by looking at an image representing the scene described in the analyzed text (Delecraz et al. 2017).

- **Jeremy Auguste** works on learning conversation-level representations. Conversations have been so far modeled either at the turn level, or globally through bag representations (bag of terms, global features). Jeremy's work consists in generating representations from hierarchical deep learning models that encompass both word-level and turn level content. Learning these representations is supervised from the content of the conversations in term of linguistic phenomena (e.g. dialog acts) as well as the external factors, such as the satisfaction of a customer in a contact center (Auguste et al. 2018).

Chapter 4

Generalization

4.1 Introduction

Generalization is the concept that a system can work in conditions that are novel compared to those for which it was designed. The first target for generalization is that a system works well on inputs it has never seen, by modeling the phenomenon it has been defined to model well. Additionally, a system that generalizes well may obtain good accuracy on domains, tasks, or types of inputs it has never seen. A system should recognize that the phenomenon it is looking at is conceptually similar to inputs it has already seen, and be able to process it. In machine learning, generalization is linked to the fact that good performance on training data does not necessarily lead to good performance on test data, for example because either training or test data are not a representative sample of the underlying distribution. It is admitted that adding model complexity leads ML systems to overtrain by "memorizing" the training data (especially when the training set is small), which decreases the expected performance on a test set. So, from that point of view, a system which generalizes well should have good performance on unseen test data. But generalization is also concerned about the scope of the underlying distribution we work with, be it narrowly tied to a specific context, or broadly defined.

Another interesting aspect linked to generalization is that systems should recognize when they are processing data that is outside the scope of their capabilities, that is inputs for which they cannot generalize. Confidence estimation and explainability are both reminiscent of this aspect and therefore strongly linked to generalization properties of a system.

In NLP, the degree of generalization of a system could be defined according to various directions of increasing difficulty. The **generalization level** of a system can be described according to whether the training data and the test data come from:

- level 1: the **same distribution** (i.e. varying the amount of training data);
- level 2: a different **time frame** (trained with past data, processing present and future data);
- level 3: a different **genre**, language register or modality (i.e. text vs speech);
- level 4: a different **topic** or specialty domain (i.e. medical vs sports);
- level 5: a different **language** (i.e. French vs Chinese);
- level 6: a different **task** (i.e. POS tagging vs semantic parsing).

These levels can also be extended to any combination of the aforementioned generalization conditions. While some of these conditions have been explored, higher levels such as language and task generalization have only recently been focused on, and seem much harder to achieve, and are argued not

to be fully reachable without general artificial intelligence (or could be interpreted as a path towards general AI).

This definition of generalization levels is not completely satisfactory as some NLP tasks may be strongly dependent on some aspects, and less dependent on others. For example, POS tagging is not very sensitive to time shifts while name entity recognition is much more; or a speech-only task such as speaker identification is not very dependent on language, although it is harder to recognize a speaker speaking a different language (Wang et al. 2012). Also, depending on the selected condition at a given level, generalization might be easier or much more difficult: historical data separated by several centuries from a source of training data is more difficult to process than texts dated a month after training data; generalizing to a subtopic of the same topic might be easier than two very different domains; a language from the same family as the one used in training data might also be easier to assess. In particular, some conditions in a given level might be harder than the conditions in a higher level. Therefore, to address these shortcomings, generalization at a given level could be recast as measuring performance in an adversarial fashion for which the most difficult condition has to be studied. This extends to whether the set of conditions is bounded and whether adversarial conditions can be described in a given setting.

Once a level is selected, given an adequate measure of performance, it is possible to measure generalization (or lack of generalization) through the drop in performance of a NLP system by fixing all conditions except for those of the target level, and varying the difficulty at that level. The literature contains many examples of studies emphasizing "adaptation" or presenting contrastive experiments on different domains, size of training data, etc. It is also worth noting that the framework for estimating a system's generalization may allow for a small quantity of training data in the target condition (often called adaptation data), or data from the general distribution from which the condition is drawn (such as leave-one-out setting on languages). Most work presented in the literature focuses on the first level of generalization, i.e. presenting performance on a test set drawn from the same distribution as training data. Also, the generalization problem is often tied to an error in experimental setup design which can be called "test distribution spill" where some factor of the test condition is present in the training data allowing a system to train on it. The classical error is to include actual test data in training, leading to spectacular performance improvements, but more subtle problems, such as using the same speaker, source, time frame, etc. in training data can result in performance gains non-representative of the generalization capabilities of a system.

We have already seen in Chapter 3 that independence assumptions may hamper the modeling power of NLP systems by not allowing them to see the range of inputs or decisions they might need to see (in a sense, generalizing on contexts which are considered independent at training is also an interesting level of analysis). This often results in mismatched conditions between training and test, and therefore a need for good generalization. Another example is training according to a loss function which is an approximation of the true loss, like in summarization where the true loss is hard to express.

This chapter is an overview of a few generalization studies, constructed around a somewhat broader setup to test generalization: systems are often built for a known target condition, without keeping performance on the source condition, but using data or linguistic knowledge from that last condition. It is organized as follows. First we look at generalizing NLP systems in the genre space, in particular to process speech. Then, we consider generalization from a task point of view, and in particular creating representations that are suitable for multiple NLP tasks, and how such representations can benefit a particular task. Finally, we look at how NLP systems can be generalized at the language level, seeking true language independence.

4.2 Generalizing Across Genres

In NLP, a genre is defined by a source of linguistic data that differs in nature. A genre is different from a domain which is more linked to applications (such as the medical domain), and is often characterized by specificities in terms of topics, terminology, and often language register. By contrast, a genre covers

much broader differences in terms of phenomena such as spontaneous versus edited, or narrative versus conversational linguistic material. In this section, we focus on the difference between text and speech, and how to generalize from the former to the latter. The degree of generalization according to the taxonomy given in 4.1 is Level 2.

Speech is one of the most efficient means of communication for us, the first we learn and probably the most natural one. Yet, the NLP community largely focuses on textual representations, such as found in email, reports, websites or news. Speech transcripts can be very different from idealized text for the following reasons:

- conversational speech is often **spontaneous**, including disfluencies such as hesitations, repetitions, revisions, etc.;
- conversational speech might involve **multiple speakers** who cooperatively build meaning: a speaker might finish the sentence of another speaker;
- multi-speaker conversations reflect the evolution of the social status and **relationship** of the speakers: convergence, divergence, competition for the floor, etc.;
- linguistic phenomena observed in text are an **idealized** or **canonicalized** version of what is observed in speech, and many phenomena, such as sentential units, are much less defined in speech;
- **para-linguistic** phenomena, such as prosody or the emotional state of the speaker, participate to the semantics of the message;
- each speaker is differently affected by the above phenomena, resulting in a varying degree of **speaker specificity**;
- automatic processing leads to **errors** due to noisy conditions, and channel effects (microphone placement or quality).

Each of these differences from clean text has led to a large body of work in the research community, specifically in the area of spoken language understanding (SLU). Even though SLU is often framed as intent classification or slot tagging tasks with application-specific tagsets, the recurrent challenge lies in training models to recognize open-domain representations, such as syntactic dependency parses, semantic role labels or semantic frames.

There are many possibilities for addressing the genre mismatch between written text and speech. The first one consists in training systems on similarly looking data, whether it comes from a similar genre or it is simulated. For example, for the ASR error detection work described in (Bechet et al. 2013[†]), we trained a dependency parser on simulated speech from text by lowercasing all the words and removing punctuation. As reported in (Kirchhoff et al. 2015), the macaon parser resulted in 10% accuracy improvement of verb morphology prediction over the (off-the-shelf) Stanford parser for a speech machine translation task. Another interesting aspect is that we trained a 1st order parser instead of a 2nd order parser because it was less affected by the snowball effect of ASR errors, and higher order would not benefit as much the short sentences of speech compared to longer sentences in text. Simulation of a target genre (as well as data augmentation such as cropping and mirroring used for computer vision) is limited by the quality of the simulation. It is well known in the community that generating texts with a language model in order to train a new language model on more data will not outperform the original language model even though LM interpolation might be beneficial (Bellegarda 2004). While vocal tract normalization (Pitz et al. 2005) is successful at simulating a neutral speaker for ASR, such invariants are harder to find in natural language processing, and simulating dialog from texts or generating paraphrases are open problems that could help genre-level generalization.

A second approach to ensure generalization of NLP systems on speech consists in including non-textual modalities in the decision. Prosody, for example, can help discriminating whether a prepositional phrase is attached to the previous or next sentence (Favre et al. 2008a[†]). It can also inform

extractive summarizers by boosting sentences with more emphasis (Xie et al. 2009a†). Yet, prosodic features are speaker-dependent and normalizing them is difficult for infrequent speakers. Using images as a source of information is also a way of better generalizing decisions on speech. In our work on speaker role recognition described in Section 3.2.1, adding image and sound representations resulted in an improvement in classification accuracy (Rouvier et al. 2015a†). For prepositional-phrase attachment resolution in multimodal context, we showed that adding image features would better resolve cases which were not well represented by text-only semantics (Delecraz et al. 2017†). The problem of extracting accurate features from image is even more prominent than for prosody: in that pp-attachment work, object detection in images suffered from a recall of 50%, which tended to cripple the benefit from derived features for the final task (due to error propagation in the pipeline).

The decision strategy implemented in a system can also increase the robustness to genre changes, especially in the context of erroneous inputs such as generated by automatic transcription. The established strategy in the spoken language processing community is to account for multiple hypotheses (ASR n-bests, word graphs, confusion networks...) as input to a system and extend them as output of the system. In the sequence tagging framework, implementing a pipeline of systems as finite state transducers, and inference as their composition, leads to the potential exploration of the complete space of combination of inputs with potential labels. While it may effectively reduce the problem of error propagation, it is affected by an explosion of the number of hypotheses, requiring clever pruning, and it assumes a joint model over the tasks which requires synchronous training data in all subtasks. In addition, even though efficient algorithms are available for hypothesis spaces expressed over sequences (regular languages), inference is harder on more general structures such as trees or graphs, and even harder for NP-complete formulations such as those used in summarization. An alternative approach is found in end-to-end training which leverages deep learning for modeling intermediate representations. In (Tafforeau et al. 2016†), we showed that the end-to-end approach was more robust to ASR errors than the pipeline approach at most word error rates. The way the problem is expressed might also influence the effect of genre mismatch on a system. In (Trione et al. 2016a†) we showed that although extractive summarization is affected by ASR errors, an alternative way to model the conversation summarization problem is to perform extraction in well-formed reference summaries, and adjust conversation-specific information to match the target conversation. The benefit comes from selecting well-formed sentence templates even though the selection process is based on erroneous transcripts. The trade-off resides in how general the gold-standard summaries are in regards to novel conversations. Conversations addressing a non-standard event, or a new topic would not be well covered by segments from the training corpus.

To summarize, even if speech is very different from text, there are a number of approaches to transfer models trained on the latter to the former: simulate or directly model invariants shared by text and speech, such as the absence of case or punctuation; include side-modality phenomena, such as prosody, video, etc.; build robust strategies such as representing multiple hypothesis, using and end-to-end approach instead of a pipeline approach, or make predictions in a space less affected by ASR errors. Reducing the problem of genre generalization to speech is, of course, naive. There are many areas where language is used, which could be called genres, and for which cross-genre generalization might be beneficial: non-verbal communication, body language, cognitive process tracking through neuro-imaging, social signals, arts…

4.3 Generalizing Input Representations

The typical input for non-multimodal NLP systems is text as a sequence of words. Due to morphology, spelling errors, proper names and speciality domains, the size of the vocabulary for a given language increases almost sublinearly with sample size. When annotated corpora are of limited size, it is indispensable for systems to generalize to new unseen words. While characters can be accounted for to build surrogate word representations, the meaning of words (or their role in a given task) is not fully compositional from characters and therefore it is often beneficial to use word types and character-

based representations as input of a system. Normalization through morphological analysis and spelling correction are another way of addressing the generality of input representations, but both are affected by non-regular words.

As we have already seen in previous sections, using multimodal cues such as prosody when processing speech, is another way of better generalizing input representations. Prosodic features are typically computed from an estimation of voicing, F0 (or pitch), energy and rhythm by measuring the timing of individual phonemes. Although a large quantity of side information is conveyed through prosody, it is hard to estimate due to the composed effect of speaker and channel. Some of that effect can be neutralized by normalizing the previously mentioned features by speaker-level statistics (assuming speaker segments are available, or using a moving window), with the median of the feature, or estimation of a baseline value with a specific model (Sönmez et al. 1997). Nonetheless, such features are often less reliable than textual input, and bring marginal performance improvements.

In this section, we look at one particular problem when using word representations: how representations can be generated at test time for unknown words. The degree of generalization according to the taxonomy given in 4.1 is Level 3 as unknown words is an issue mostly observed when changing domains or topics.

Word embeddings are compact vector representations for words that place words with the same properties close together in the embedding space. Depending on how they have been trained, they can reflect syntactic, semantic or topical proximity. Embeddings based on the distributional semantics hypothesis (Harris 1954), which assumes that word meaning can be defined by which words they tend to cooccur with, depending on their training regime, will lead to different kinds of representations. The most popular techniques, CBOW and skip-gram (Mikolov et al. 2013), or GloVe (Pennington et al. 2014), are reminiscent of Latent Semantic Analysis (Deerwester et al. 1990) and correspond to a low-rank version of the cooccurrence matrix which counts the number of times two words cooccur in a given text window. For example, GloVe explicitly trains representations so that the scalar product between vectors representing two words is close to the normalized log-cooccurrence of the pair. Even though word embeddings can be trained on very large corpora (several billion words), devising representations for out-of-vocabulary words can be difficult.

Word embeddings can be used as is as features of an NLP systems, or they can initialize the embedding layers of a neural network and be fine-tuned for a target task. Therefore, it is reasonable to assume different types of OOVs:

1. words without a representation because they were not part of the embedding training corpus, and not part of the task-specific training corpus;

2. words which have a representation but are not present in the task-specific training corpus;

3. words which do not have a representation but are present in the task-specific training corpus.

In the simple case where no fine-tuning is performed (cases 1 and 3), rare words can be replaced by a special $\langle unk \rangle$ symbol which corresponds to the 0 vector that cancels the transformation applied to the input in the first layer of the embedding layer, and therefore a representation is completely devised from the bias associated to that layer. At test time, those words, such as proper names, can be reasonably handled with a single class of OOVs (case 1). However, for case (3), it might be problematic to collapse relatively frequent tokens for which representations are unavailable to a single $\langle unk \rangle$ representation. For example, tokenization of frequent multiword expressions such as particle verbs, set phrases, might induce OOVs which would benefit from a specific representation. Therefore, it is always advised to retokenize the embedding training data with the same rules that pertain to the tokenization of the task-learning corpus. Using approximate match or composition from known token representation is an option but it remains heuristic.

Fine-tuning embedding layers initialized with pre-trained representations induces another problem. Case 3 OOVs (in training data but no pre-trained representation) are not a problem anymore because the system can learn a representation for them as long as their initialization is random and they are not

collapsed to 0. If rare enough, words which have no representation and are not in training data can be collapsed to a single $\langle unk \rangle$ symbol[1]. Words which have an embedding but are not seen in training data (case 2) are problematic because the fine-tuning process might have transformed the embedding space by a large-enough amount for them to induce wrong classification decisions. We have shown that given a small text sample containing an OOV, it is possible to compute a distribution similarity with other words of the lexicon, and borrow their embeddings (or an interpolation of embeddings from multiple close neighbors) and use them before or after finetuning resulting in performance improvement on a POS tagging task (Tafforeau et al. 2015a[†]).

More recent work, such as Fasttext embeddings (Joulin et al. 2016), train jointly skip-gram model for word forms and character n-grams (morphological units). If a representation is not available for an OOV, then it can be devised from morphological units. This type of representation is very successful at handling morphology-rich languages, but it fails to generalize for words which have uninformative morphology but follow distinguished distributional patterns. Another trend is to build contextual embeddings, that are not fixed for a given word, but recomputed from the context it occurs in (Peters et al. 2018). The principle of ELMo, embeddings from language models, is to train a multilayer bidirectional RNN (such as Bi-LSTMs) on a language modeling task (predict the current word given neighboring words) on a very large set of texts. The embeddings are the activations of the network before the decision layer. To use those embeddings for an NLP system, one must process its training and test data to collect the corresponding activations and use them instead of regular fixed word embeddings. The contextualization of the representations has been shown to improve on a range of tasks, probably due to the capacity to adjust representation to the actual sense used in a context. The ELMo implementation also addresses the problem of OOV by using a character CNN layer which models basic word representations with character-level convolutions (similar to n-grams), a fairly standard practice in recent systems. One should note that the ELMo approach could be generalized to any task, and in particular it has been known for some time in the deep learning community that the first few layers of a system should benefit from pre-training on large corpora on an "unsupervised" task, such as auto-encoding (Vincent et al. 2010), which is effectively what ELMo does. The question is that what tasks having large quantities of training data might be beneficial to the task of interest for which the system is pre-trained, and are there generic pre-training tasks, beyond language modeling, which would be beneficial for any kind of end task?

Even though it is possible to devise representations for out-of-vocabulary words from their morphology or the context in which they appear, there are three cases for which word embeddings do not generalize well: small corpora, novel senses, and new languages. A rule of thumb is that a corpus of roughly 10 million words is necessary for training reasonable embeddings, and adaptation methods are necessary to leverage existing embeddings. Many methods have been proposed for training sense-aware embeddings (Qiu et al. 2016; Li et al. 2015; Chen et al. 2015), yet they either assume a set of topics (under the assumption that words have a single sense in a given specialty domain) or a set of senses, and cannot account for new senses. What makes novel word senses emerge and how much data is needed for evidencing this emergence remains an open question (Hamilton et al. 2016) even though changes can be evidenced after the fact[2]. In the context of domain adaptation, this is again difficult. Those problems can be extended to the training of multilingual embeddings which, assuming bijective translation, can be used to train language-neutral systems and therefore generalize across languages. The dream of universal embeddings is an elusive one because an expression might be translated to a single word in a different language, and because some syntactic words have no translation in other languages.

In addition, the meaning of expressions can be captured from individual word representations by

[1] Alternatively, multiple symbols such as POS tags could be used to represent OOVs as long as their distribution in training data is representative of their occurrence in test, since the corresponding representations, randomly initialized, can be trained towards the target task.

[2] In unpublished work, we showed with A. R. Balamurali that aligning embeddings trained on temporally disjoint corpora would result in a set of words being far apart compared to the general population, evidencing a change of sense for those words.

compositional models such as RNNs, but how they process non-compositional meaning is still not very clear although we know that coocurrence models such as word2vec are able to capture some of the meaning of non-compositional expressions if they are trained at that level of granularity. How do these phenomena combine and how they match the degrees of non-compositionality processed by humans remains an open question.

4.4 Generalizing Across Tasks

Often, when performing NLP tasks, we look at a single problem. For instance, parsers are trained on treebanks for the purpose of parsing, a text genre classification system is trained on a categorization dataset, etc. Hypothesizing that tasks are independent from one another is compelling because the user is only interested in a particular task, training data may be annotated for a single task (treebanks may not be annotated for genre), the structure of the problem might be specific to the task (predicting trees for parsing, genres for categorization), etc. Yet, there are cases for which tasks share common factors which could be leveraged. A well-known example is the benefit from performing speaker adaptation in automatic speech recognition (Leggetter et al. 1995); (Rouvier et al. 2014a[†]), or topic adaptation for the same task (Lecorvé et al. 2008). Although speaker identity or topic are not the end problem, making predictions (or generating representations conveying those factors) is beneficial to the task. For the example of parsing and genre categorization, it is known that written text yields a different distribution of syntactic phenomena than spoken text, starting with sentence length and punctuation distribution, therefore a genre categorization system might benefit from syntactic features, and a parser targeting a multi-genre domain might benefit from a genre feature. This is true for many language-related tasks, and especially at the task composition level: sentence segmentation may benefit from parsing (Favre et al. 2008a[†]) and (for obvious reasons) conversely; parsing may benefit from word sense disambiguation (Agirre et al. 2008) and conversely (Navigli 2009); word embedding training (a variant on the language modeling task) may benefit from parsing (Levy et al. 2014) and parsing may benefit from pretrained word embeddings (Chen et al. 2014). There are so many studies of the co-benefit for NLP tasks that it would be unreasonable to list them all. Yet, truly joint models for non-trivial tasks have only been flourishing recently thanks to advances in deep-learning. Even though early attempts looked at joint optimization techniques such as dual decomposition (Le Roux et al. 2014), the turning point occurred with the "NLP from scratch" work (Collobert et al. 2011) which advocates sharing model architecture and weights across tasks.

The work described in this section considers a degree of generalization of Level 6 according to the taxonomy in 4.1. However, we do not consider an unknown task but rather share model subsets among multiple tasks in order to improve generalization on a target task.

In (Tafforeau et al. 2016[†]), we proposed to predict Framenet labels from noisy ASR transcripts using a multitask system. Compared to a pipeline approach, the two motivations of that work are that the ASR output generates noisy text which impacts decisions in the pipeline and precludes robust results, suggesting that an end-to-end approach might be less prone to errors, and that linguistically relevant intermediate representations from the pipeline drive generalization and should still be accounted for when training the system. Therefore, we create a system that would jointly predict part-of-speech tags, disfluency markers, named entities, dependency parses and finally framenet parses, all being predicted from the input words. The first tasks were considered as auxiliary tasks supposed to help the frame generation system generalize, by learning intermediate representations useful for all tasks without relying on actual symbolic representations of the linguistic phenomena, and therefore being less prone to input and pipeline errors.

The problem was set up as a word-synchronous prediction problem where for each word a vector of symbols are predicted with one symbol per task. While part-of-speech tagging was naturally represented in this framework, the other tasks required adjustment: disfluency and named entity segmental predictions were cast as begin-inside-outside predictions (which allows recovering the original labeling), and dependency parsing was degraded to labeling each word with the dependency link label

and relative index to its governor (which is hard to generalize for long-range dependencies). Frame parsing was also cast as a begin-inside-outside tagging task with specific segment categories for the frame, the trigger and the frame elements (assuming a one-frame-per-word hypothesis which was true on the corpus, but is not true in general). The predictor was a bidirectional LSTM inputing word embeddings and outputting task-general representations which were then fed to task-specific decisions layers. This system is equivalent to building a separate model for each task and then sharing the weights of all layers up to the task-general representation generating layer. The system is then trained by minimizing the sum of the losses for all of the tasks.

One problem with synchronous joint training is that some tasks might be easier than others or some tasks might be less important than others (i.e. auxiliary tasks) yet they all contribute the same to the loss function. We have shown in this work that, in addition to outperforming the pipeline and single task approaches, the losses of the auxiliary tasks could be weighted non-uniformly in order to increase performance on the main task. The system also turned out to be more robust to ASR errors across a range values of WER. Some questions remain on how to choose task weights (are they another hyperparameter?) and how to size the number of parameters of the single task systems versus the multitask system in order to get fair comparisons.

Multitask learning is a great way of incorporating co-dependencies across tasks in representations. Even though we explored synchronous multitask settings, it is possible to train each task separately on different corpora, sparing the annotation cost of building ground truth for joint training. Multitask learning has seem a lot of developments in the machine learning community with generative adversarial networks (Goodfellow et al. 2014; Gulrajani et al. 2017) which joint train a system to generate samples from a distribution and to discriminate between human-generated and machine-generated samples, and in reinforcement learning where auxiliary tasks can help learning hard to explore long-range objectives (Jaderberg et al. 2016). An interesting question was raised by researchers exploring the idea of building a single neural network for solving all problems (not necessarily related tasks): as tasks are switched, a system tends to forget what it learned when confronted to previous tasks; how can we prevent this forgetting from happening (Lopez-Paz et al. 2017)?

It is noteworthy that word embedding pretraining is another kind of task generalization (Level 6). Word representations are trained on a language modeling task (predicting the cooccurrence of two words, predicting a word given its context…) but used for a different task, such as semantic parsing. The point is that sharing a subset of parameters across tasks has proved to be beneficial but sharing the whole model (including input and output representations) is much more challenging and would probably fail to generalize to novel tasks in a zero-shot setting (McCann et al. 2018). Howard et al. (2018) give high-level approaches for fine-tuning a general language model to obtain a multitask system by using a combination of layer-specific learning rate schedules, and integration of contextual representations through pooling. The Generative Pre-training Transformer (GPT) approach extends those representations by training a language model that builds prediction on multiple layers of "transformer" blocks which are essentially multi-head attention mechanisms. Then, this architecture is pre-trained as a language model on a large corpus, and fine-tuned with an addition projection layer for specific tasks. It can be used by encoding the input as a sequence of words, for tasks such as text classification, textual inference, or multiple-choice questions, and use a task-specific linear classifier for generating decisions. In order to improve generalization, other auxiliary tasks can be devised, such as a masked language model (where some input words are masked and must be predicted), or next-sentence predictor (whether two sentences are consecutive or not), which yields very good unsupervised representations on top of which simple task-specific classifiers can achieve good performance on many tasks (Radford et al. 2019). Generative tasks, such as machine translation or summarization can be tackled this way by using the representation to rescore a word-sequence generator[3].

[3]Maximizing prediction score in that framework is, however, very expensive.

4.5 Generalizing Across Languages

Some argue that machine-learning based NLP systems are language independent because they are functions with free parameters that can be retrained on a new language or domain. It is true to the extent that assumptions on the language or language family the system was trained on hold for the target language. For example, assuming an easy tokenization based on spaces and punctuation works for western languages but fails on Sino-Tibetan languages which do not have spaces. Conversely, sentence segmentation is trivial in these languages because they have dedicated punctuation (only since the 20th century) while it is a harder matter in languages with ambiguous punctuation. Another example is rich morphology which can greatly increase the number of word types in a language, rendering token-based modeling less useful, and which often comes with less constrained word order that renders constituency modeling of syntax not as effective as dependency-based modeling. In ASR, languages which have a high degree of heterographic homophony such as French are more difficult to process than languages with low level of phoneme-to-grapheme ambiguity such as Spanish.

Most NLP research was and is still driven by the English-speaking community and correspondingly most empirical evidence in NLP-related communications is presented on English reference corpora first, and later on other languages. The machine learning community being less interested in linguistics, this bias will get stronger, with results presented on old reference corpora such as the Penn Treebank which have been over-fitted for years. Recent efforts emphasize language independence by bridging annotation formalisms across languages, and creating coherent resources which span many languages.

The American agencies such as DARPA and IARPA were the first to push multilingualism for strategic reasons. For example, the DARPA GALE or BOLT programs systematically targeted non-English languages such as Chinese and various dialects of Arabic. However, until recently the best strategy for processing a new language was collecting data in that language and retraining or fine-tuning a model on that data. The only option for truly fostering research on language-independent NLP is to cast the problem as one where no training data is available on the target language although some is available in potentially many languages different from the target language.

The Multiling summarization evaluation series is a community-organized research initiative which focuses on truly language independent NLP by providing summarization evaluation data in 40 languages for single document summarization. Featured articles from Wikipedia and their translation in diverse languages are provided to the participants who have to automatically generate summaries with a single method. The absence of large quantities of training data precludes the use of supervision, and every subtask necessary for extractive summarization, such as tokenization and sentence splitting, become hard. In 2011, we participated to the TAC pilot track which became later Multiling with a system focused on removing language-dependent assumptions in sentence splitting, tokenisation, and stop-word lists, within the framework of a baseline extractive summarizer (Hmida et al. 2011†). For sentence splitting, we relied on the crude heuristic that all languages have punctuation and the last character of the input must be a sentence-ending character, and used that character to devise splitting points. Existing "language-independent" sentence splitters, such as Punkt (Kiss et al. 2006) could not be used because they assume knowledge of punctuation characters, and availability of large corpora. For addressing tokenization, our system relied on character n-grams (with n=5 determined on a development set). Frequent words such as determiners and prepositions are typically ignored when computing the relevance of a sentence, hence the need for stopword lists to be able to ignore them. With high-order n-grams, those words tend to be ignored or covered as part of expressions, and no stop-list is needed anymore. Results in terms of ROUGE evaluation show that the language-independent system is not worse than a language-dependent baseline which uses language-specific resources (sentence splitter, tokenizer, stop lists) and even better in some cases (Hmida et al. 2011†). As an extension, it would be interesting to test if those results hold with more recent evaluation data covering more languages, and with other sentence selection algorithms. Performing abstractive (which generates new text) instead of extractive (which selects sentences) summarization in a language-independent framework is an open problem which would be interesting to explore in the context of the recent development of abstractive summarizers built on the deep learning encoder-decoder framework (See et al. 2017).

Another great effort for multilingual NLP is the Universal Dependencies (UD) initiative which aims at building a unique representation of syntax (including part-of-speech tags, morphological features, dependency tree and label structure), by converting existing resources such as the French Treebank (Nivre et al. 2016) to the UD standard and annotating new resources. UD is an ongoing effort gathering more than 200 contributors, 100 treebanks on 70 languages[4]. The CoNLL2017 and 2018 shared tasks have focused on building parsers for the UD data but organizers seem to be less interested in the idea of evaluating language independence on unseen languages. However, in the 2017 edition, there were a small number of surprise languages which had little training data (only enough for a development set), and a number of participants have explored training on close languages, or training delexicalized parsers (parsers which only see POS tags and morphological features as input) as a means of processing such challenging cases.

"Zero-shot" syntactic parsing, which means training a parser on a set of languages and testing it on an unseen language, is mainly motivated by the cost of building treebanks for uncovered languages. It has been addressed in the past with two main approaches. Unsupervised parsing consists in learning a grammar from statistical regularities in a corpus of sentences annotated with POS tags (Sankaran 2010; Bod 2007; Naseem et al. 2010; Jiang et al. 2016). It often relies on an application of the EM algorithm which infers a labeling of the data with an initial model, reestimates that model from the inferred labeling, and loops over the process until convergence. Even though they only require unlabeled data, those approaches are computationally expensive on larger datasets, yield relatively low accuracy compared to supervised parsing, and the produced analyses are often hard to interpret. The second category of approaches is based on transfer learning which uses a full parser on a close language and then translates the source language automatically so that it matches the properties of the target language (Lynn et al. 2014). The transfer can be operated through various machine translation techniques, such as transliterating words. Those two categories of approaches suffer from not accounting for the large number of annotated treebanks in a variety of languages, such as available in UD.

Recent trends, fostered by the aforementioned CoNLL shared tasks, consist in building parsers from the concatenation of all UD languages in order to parse unseen languages. Multiple challenges need to be accounted for:

- Pre-processing such as sentence segmentation and tokenization need to be language independent. We have already discussed that matter in the framework of summarization but real tokens are needed here, and unsupervised word segmentation techniques or a general sequence tagging model could be used to infer word starts and ends from characters (Cai et al. 2017).

- Word representations need to be language-independent, by using for instance multilingual word embeddings (Duong et al. 2017; Artetxe et al. 2017; Conneau et al. 2017), which can be trained independently on all languages, and mapped together by learning or using a given bilingual dictionary[5]. However, the challenge is that this approach assumes a bijection between the lexicons of each language pair, which is a crude approximation of reality. An interesting question is how to deal with multiword expressions which are often non-compositional.

- Recognize and leverage linguistic invariants such as word order, and markers at the token of morphological level (for instance plural, gender, genitive…). Such invariants are described in resources such as the World Atlas of Linguistic Structures (Comrie et al. 2005) for a large number of languages, and could inform a parser when learning from many languages (Ammar et al. 2016a).

- Account for the noise and distributional specificities in UD. UD is the concatenation of many treebanks built independently from different data sources, with different rules, annotator experience, etc. The normalization process is often automatized with corner cases which are difficult to tackle. In addition, the level of coverage for morphological features varies across languages.

[4] As of 2019 (https://universaldependencies.org/).
[5] Such as PanLex https://panlex.org.

In (Scholivet et al. 2019†), we have shown that multilingual generalization of a syntactic parser, the discovery of multilingual invariants during training on a concatenation of UD corpora, was not as straightforward as it seemed. Linguistics has long sought universals in the use of language by humans. It is hypothesized that an underlying parametrable grammar exists, and that each language corresponds to a configuration of that grammar (Greenberg 1963; Chomsky 1995). Under that assumption, we trained a delexicalized parser (in which words are replaced with part-of-speech categories augmented with morphological features) augmented with language-level features supposed to inform the parser on languages sharing properties such as word order.

The parser is a standard shift-reduce feed-forward neural network which predicts parsing actions given a configuration of the automaton. It is trained on the concatenation of balanced UD annotations for 40 languages. We compared a baseline system with a system augmented with a vector 22 language-level features from the WALS (Comrie et al. 2005). Results show that the performance of the augmented model is significantly better than that of the baseline model by 4 points of LAS on average. An interpretation of this result is that linguistic configuration features are indeed improving the generalization of the parser. However, and alternative interpretation could be that the parameter space of the parser is just split among languages, and that the system first predicts the identity of the language and then uses that identity in order to select the subspace of parameters which correspond to that language.

In order to untangle those two interpretations, we trained a parser augmented with a single feature which identifies the language. Its performance is significantly lower than of the parser trained WALS features, but better than the baseline model. This tends to indicate that there is some truth in both interpretations. Furthermore, a simple language classifier trained from the neural activations of the parser's MLP show that language identity can be predicted with very high accuracy when the WALS features are present, while basic parser configuration features do not allow such prediction. This use case shows that generalization is not just a matter of giving the right inputs to a system, it is also linked to the topology of learned parameter space. How to efficiently use conditioning such as linguistic knowledge for helping systems generalize remains an open problem.

Even though the issues associated with training language-independent parsers are challenging, there are ways of casting the problem which might lead to breakthroughs. For example, one could imagine a setup where a system needs to be independent of the alphabet of the input. One simple way of testing the generalization of such system would be to create a permutation of the alphabet of a language used in training, and apply it to the test data. The system would have to recover its original performance on that permuted dataset. While at first glance, a typical deep learning system (for POS tagging for instance) would fail catastrophically because it assumes the identity of characters to remain at test time, one could try to interpose a transformation between the input and the first (character embedding) layer of the network, and seek the transformation that best matches the distribution of activations in the network on an unmodified test set.

Formally, let X be the distribution of original inputs, $X' = PX$ is the distribution of permuted inputs according to an unknown permutation P. Let $N(X)$ be the distribution of activations over the original input space. We can apply to X' the transformation T as TX' and try to recover T which minimizes a distance over distributions (such as KL) between the activations of the original inputs $N(X)$ and the transformed permuted inputs $N(TX')$.

$$\hat{T} = \underset{T}{\operatorname{argmin}}\, D_{KL}\left(N(X)||N(TX')\right) \tag{4.1}$$

The distance should be minimal when $\hat{T} = P^{-1}$ since $X = P^{-1}PX = P^{-1}X' = TX'$. The approach described here is very similar to the approach used to map two word embedding spaces from different languages so that words which are translations of each other lie in the same place (Conneau et al. 2017). Unless you are a cryptographer, a permutation over the alphabet is a naive way of thinking of the difference between two languages. However, the idea could be extended to transformations on non-trivial parts of the input such as character n-grams which can encode morphological phenomena.

4.6 Conclusion

Generalization in NLP is very important to the community, as evidenced by the NAACL 2018 workshop on New Forms of Generalization in Deep Learning and Natural Language Processing which tackled the questions of measuring generalization, and building systems that generalize better. The conclusions of that workshop were that more inductive bias should be used so that models can better capture linguistic invariants, that systems should make use of common sense reasoning in order to better generalize, and that they should be tested on unseen distributions and tasks[6]. Those conclusions are in line with the arguments developed here and the study of generalization is still a mostly unexplored area of NLP.

We have seen in this chapter that for a range of generalization levels, given a good understanding of the properties of the distributional shift studied, specific systems could be created that are able to address some of the challenges associated with adversarial test data. However, composing generalization conditions increases the difficulty and it is not clear how the human expertise which was efficient on single conditions can be automatically composed to generalize well on those conditions. Recent research looks at how visual deep learning systems can generalize to new attribute pair or triples in tests similar to IQ tests (Barrett et al. 2018) and show that comparing instances is critical to capturing higher-level invariants. This observation probably means that kernel methods, which build on measures of similarity between examples, will probably play an interesting role to tackle the problem. Nevertheless, the theoretical question behind such conditions is whether general artificial intelligence needs to be solved before they can be approached.

In addition to studying the composition of distributional shifts, there are other very attractive avenues of research that are linked to generalization. The first one is how we can generalize across modalities. Can invariants learned on a visual task be exploited to process an NLP task? Can problem characteristics learned on a NLP task be accounted for to process a physical haptic problem (in the context of a robot). Humans can perform such generalizations, and systems should be able to do so as well. The benefit of studying this type of problems is that it should also help us build a better model of how the human brain processes multimodal information on how sensory information is converted into linguistic assertions, potentially tackling the frontier between thought and sensory input.

Another prospect is to leverage user interactions to enhance system generalization. The closed environment induced by scientific methodology tends to remove the user from the loop, to ensure repeatable research, but there are large potential gains by allowing the user to interact with a system. The user could input some adversarial training examples in an active learning loop, or could use interactions to discover system weaknesses. This area of research is still unpaved and requires much more effort than the traditional experimental setup which excludes the user.

To conclude this chapter, before building more general systems, the challenge to come is to build adequate methodological artifacts to show the good generalization of NLP systems, and as we will see in the next chapter, it is not evident that current methodology is suitable even when addressing the first level of generalization.

PhD student work relevant to this chapter

- **Jérémy Trione** worked on abstractive and extractive summarization of speech conversations from call centers. One particular problem with the extractive approach for speech summarization is that there is a mismatch between the source content (errorful automatic speech transcripts), and the target content (fluent narratives). Jérémy proposed a method for transferring between those genres in a scarce data scenario by extracting template sentences from a set of reference summaries for a training set and patching them with information from the target conversation (Trione et al. 2016a).

[6] Gen-Deep18: https://newgeneralization.github.io/. A good summary of the workshop can be found here: https://thegradient.pub/frontiers-of-generalization-in-natural-language-processing/

- **Manon Scholivet** works on language-independent natural language processing. In particular, she has been interested in training syntactic parsers that can learn from a set of languages and generalize to unseen languages. In particular, she questions whether deep methods can recognize and leverage linguistic invariants across languages when concatenating training data from multiple languages. First results have shown that deep models tend to isolate languages, resulting in sub-par generalization with regard to what would be expected (Scholivet et al. 2019).
- **Thibault Magallon** works on contextual inference for optical character recognition. In particular, he proposed a method for detecting and correcting errors in OCR systems output, and is exploring models that go beyond the local visual context typically tackled by OCR systems, leveraging non-local CNN representations of pixels and text hypotheses (Magallon et al. 2018a).

Chapter 5

Methodological challenges in Natural Language Processing

5.1 Introduction

How do we prove the success of an approach in natural language processing? In chapter 2, we have seen that the most commonly adopted approach is to create annotated corpora and benchmark our systems according to an evaluation metric on these corpora. We make sure that such evaluation is the most rigorous, by using appropriate test splits, meaningful evaluation metrics and significance testing. If sufficient effort is put into documenting the process[1], the work presented in articles might be reproducible (given data and code, obtain the same results), but does it make it replicable (if we were to recollect the corpus, would we obtain the same result)? When scientists compete for research funding, there is an inevitable bias towards publishing the most exciting results, at the cost, sometimes, of fraudulent behavior[2]. What can we do to preclude such behavior?

The community has resorted to shared tasks, which separate the evaluator from the evaluee, providing independent judgments on newly collected data. Although there is a certain interest to organizing such evaluation campaigns, do they present all the expected benefit? Another important aspect is the objective function of the field: improve performance according to a specific metric that assesses system quality. What are the choices associated with selecting a particular metric, and how do they impact the results of research? Can we build more relevant metrics by extending the scope of what they evaluate? What are the compromises associated with such practices?

This chapter relates a few aspects linked to my research in the methodological areas of NLP. Section 5.2 reports how the community uses shared tasks as a way of obtaining reliable estimates of progress at a task scale. It outlines some of the problems linked to that approach and discusses a few potential solutions. Section 5.3 focuses on evaluation metrics that account for how end users of a system perform their own task with its help. It discusses that topic under the umbrella of two studies and tries to outline some of the challenges and benefits of that approach.

5.2 Independent Evaluation Through Shared Tasks

Natural language processing is a discipline balanced between science and engineering in which non-theoretical advances are supported by empirical evaluation, and extensive comparison. The standard setup to engage with a scientific hypothesis is to define a test set, annotate it with gold standard

[1] By making data, annotations, source code available.

[2] Scientists may fake data to obtain more compelling results (Simonsohn 2013), or may interfere with the review process leading to spurious article acceptance (Haug 2015). Our community is even interested in the linguistic aspects of fraudulent behavior (Markowitz et al. 2016; Braud et al. 2017).

labels and compare the performance of a system on that test set according to various setups. In order for test results to be unbiased, parameters and hyper-parameters of the system shall not be tuned according to performance on the test set. For comparison purposes, one will often reimplement previous work systems or quote previous work results on the same test set. This leads to two biases which may undermine the breath of scientific findings: the lesser effort put in creating the baseline can lead to underestimated baseline performance, and reusing test sets may lead to system selection, and test-informed hyper-parameter tuning.

The first effect is due to the engineering effort required to build strong baselines, and is often exceeded by that put in building the system supporting the hypothesis. The second effect is well documented as evidenced thereafter. In the artificial vision field, datasets such as MNIST or CIFAR have been overfitted for several years. A recent study has shown that on newly collected data, a range of object identification systems perform worse by 4 to 10 percentage points (Recht et al. 2018) compared to the figures reported on the original corpora test sets. In the NLP community, the ATIS dataset is used for benchmarking Spoken Language Understanding systems (Price 1990). Although many benchmark papers are calibrated on this dataset, it has been shown that it may be too simple for requiring fancy systems, and that above 95% accuracy the remaining errors are mostly annotation errors (Béchet et al. 2018). When training POS taggers on the Penn Treebank (Manning 2011) reached a similar conclusion that remaining errors above 97.3% accuracy were manly due to annotation choices. So it seems important to periodically produce new test sets in order to prevent overfitting old corpora. It also seems equally important to create reference points (by evaluating on well-understood tasks/corpora) for comparing very novel approaches to well-established approaches in order to better understand the benefits of novelty.

Shared tasks, also called scientific competitions, have been organized for decades in the NLP community in order to decouple system creation from evaluation. They guarantee that a system was not tuned on a test set by building a new test set at each iteration, and allow multiple participants to submit results generated by systems showcasing their respective scientific hypotheses. It is admitted that shared tasks remove both the weak baseline and the test set reuse evaluation biases because groups who compete put most effort in their approaches, and the test set is novel. Therefore they provide a snapshot of the state of research on a particular task at a given time.

I participated in many shared tasks, as a system designer or as an organizer. In 2005, I participated in collaboration with F. Bechet to the ESTER named entity recognition in speech competition (Favre et al. 2005a†; 2005b†), and more recently in 2012 to the ETAPE structured NER task. These evaluation campaigns featured a post-evaluation adjudication phase in which participants were tasked to improve the gold standard of the test set. This lead to a lot of discussions, in particular over the 2012 results, which were not always beneficial to the competition. In 2006-2009, and 2011, I participated to the DUC/TAC summarization shared task which consisted in producing 100 word summaries from a set of news articles (Favre et al. 2006a†; Favre et al. 2007b†; Gillick et al. 2008†; Gillick et al. 2009b†; Hmida et al. 2011†). My 2008 and 2009 entries, conjointly engineered with D. Gillick, B. Bohnet, D. Hakkani-Tür and Y. Liu were very successful thanks to the modeling of summarization as a budgeted set cover problem of word bigrams over sentences, while the trend at the time was to learn sentence-level scores. The model turned out to be successful for three reasons: it simulated somewhat accurately the scoring unit in ROUGE evaluations, it focused on readability with a trained sentence splitter and parsimonious rules for sentence compression candidate generation, and was tuned extensively to optimize for the ROUGE metric. In 2009 and 2010, I participated to the GREC referential expression generation shared task (Favre et al. 2009a†; Favre et al. 2010b†). Only few participants from the natural language generation community were interested in this task, and the success of my participation resulted from framing the problem as a sequence prediction problem and train a CRF to perform the task. In 2007-2009, and 2012-2015 I was involved with the DARPA GALE and BOLT programs under the SRI lead teams, and participated to the evaluations organized through NIST on machine translation. For those, I was only in charge of a small sub-problem, and the challenge was more in building a coherent system from multiple pieces, and training it towards a global objective with always changing bricks. In 2012-2014, I was part of the PERCOL team for the REPERE person identification in videos

challenge (Favre et al. 2013a†; Bechet et al. 2014a†). It turned out to be a very interesting competition and our success was mainly due to bypassing the direct objective of the task[3] by identifying the context in which certain categories of people would occur, and specializing systems according to the TV show they were targeting. It defeated the aim of the shared task in that the proposed approach would not generalize well to TV shows not seen in training. This resulted in our team underperforming at the related evaluation campaign from MEDIAEVAL 2015 (Bendris et al. 2015†). In 2015 and 2016, I collaborated with M. Rouvier while he participated to the DEFT, TASS and Semeval sentiment analysis campaigns, designing a very competitive and often winner system based on multiple representation learning strategies (Rouvier et al. 2015c†; Rouvier et al. 2016a†).

My take on these shared-task experiences is that they require a lot of engineering effort, often beyond what a team of researchers can be expected to do given the constraints, and that although innovation is the key to success, bugs can undermine the success of methods which are otherwise quite sound. Even though all participants are subject to the same constraints and issues, it is hard to untangle the appropriateness of an approach to solve a problem, and the related engineering issues.

As an organizer, I first proposed a meeting transcript summarization track at TAC 2010 which was supported by the community but did not happen due to the cost of manual evaluation. I then helped with the annotation and evaluation effort of the Multiling summarization evaluation shared task in 2011 and 2013 (Giannakopoulos et al. 2011†), and organized its meeting summarization pilot track in 2015 (Favre et al. 2015†) in the framework of an EU-funded project. These experiences were very interesting and discussion with organizers of other shared tasks show that beyond typical organizational problems, the issues encountered are similar to that of participating to a shared task: bugs in dataset preparation or evaluation scripts can delay and undermine the results of evaluation campaigns.

Although shared tasks are a true benefit for the community, they are subject to a number of problems. The first and most important problem is that shared tasks are a competition and the incentive to rank well can harm the scientific benefit. It puts a lot of emphasis on engineering the approach, which is driven by performance, not necessarily on explainability, and therefore may generate systems which work well but are not of great scientific value to the community. DARPA machine translation evaluation campaigns or Kaggle competitions are a good example in which winning entries end up being combinations of many systems. For example, it is not clear how Netflix benefited from the winner entry to their prize which was a combination of many systems (Bell et al. 2007; Bennett et al. 2007). Model ensembles, the combination of different parameter optimization results for the same model (Schapire et al. 1999; Zhou et al. 2002b), have also been shown to outperform single optimization runs in recent work (Jozefowicz et al. 2016), which in a competition puts more emphasis on computation power than scientific results. Another aspect often encountered in the NLP community is the effect of corpora on participating systems. Many competitions create different tracks for open and closed resources, which correspond to allowing the participants to leverage any resource such as corpora or lexicons, or limiting the set of authorized resources so that every participant uses the same input. Although this approach is a step in the right direction, it does not preclude the use of previous experience, such as the choice of heuristics or initial conditions when designing systems. Imposing a computational environment, as in the TIRA evaluation as a service initiative (Gollub et al. 2012), could limit the effect of resources on results and force participants to open-source their systems, and combined with guidelines on prior knowledge usage, might improve the outcome of shared tasks. At least it enforces the reproducibility of research, but it comes at a higher cost for organizers and engineering cost for participants (because they do not choose the computation platform). Another benefit of packaging systems as stand-alone repeatable runtime bundles is that from year to year the same systems can be rerun on newly collected data at a very low human cost, resulting in stable comparisons regardless of the availability of participants.

The second problem is that by having many research groups compete towards the same objective, shared tasks tend to reduce diversity. In order to monitor progress, shared tasks are often repeated multiple years in a row, and one of the best strategies for succeeding is to study participants systems

[3]Modeling the identity of a person in videos.

from previous years and replicate the ideas of the top participants. This effect was prominent in the TAC evaluations where most participants would implement variants of extractive methods, resulting in little significant differences between systems (Dang et al. 2008). Ensuring diversity could be achieved by organizing shared tasks for as many scientific problems as possible, and this approach is encouraged by the European Commission and Agence nationale de la recheche (ANR) in the projects they fund. Nevertheless, it is hard to find funding for recurring events outside of typical project duration of 3–5 years (for example, Multiling is organized every two years, and is self-funded by participants), and the more shared tasks are organized, the less research groups are available to participate. The additional work represented by the participation to a scientific competition is significant and has to be balanced against other research activities. The problem of diversity is hard to address without a global view on the inter winded effect of the numerous shared tasks on research groups. Maybe a central international organization body, such as the role played at some occasions by NIST in the USA or LNE in France, would be an interesting option to explore?

5.3 End-User Evaluation

The quality of a system can be measured by estimating its performance on a set of tasks. Generally speaking, there is a tradeoff between performing well on a particular task and performing well on many tasks. The choice of an evaluation metric is therefore what drives the choice of system designers along that tradeoff. As outlined in Section 2.4, there are many ways of evaluating system performance, and even on a particular task, multiple evaluation metrics are possible. However, all metrics presented in that section have in common that they compare a system output to a gold-standard behavior, be it defined through annotations, or through humans' preferences. Eventually, we are not interested in the performance of NLP systems, what we generally care about is whether those systems solve human problems or help solve them.

In that context, the best way to evaluate a system is probably to measure how helpful it is in a real, meaningful task. This means that instead of measuring a system's performance, we should evaluate the performance of humans using this system to perform their own tasks, tasks that are meaningful to them. This outlines an evaluation framework where humans are given a realistic task, that they would perform during their work hours for instance, and evaluation consists in computing the difference between how they perform when they use the system, versus when they don't, or how they perform given two systems. This kind of evaluation, called end-user or utility evaluation, is better than intrinsic evaluation (measure the performance of a component of the whole system) and downstream evaluation (measure the impact of that component on the performance of the whole system) because it directly measures the utility of the system for humans.

5.3.1 Evaluating utility

There are a number of challenges posed to construct end-user evaluation frameworks:

- the ecological validity of the task;
- the choice of a metric to evaluate subject performance;
- the definition of the baseline condition;
- the control of variability;
- the impact of the system on the task.

An ecologically valid setup is an experimental setup where the participants are not subject to any artificial constraint, and perform a task meaningful to their objectives. Ecological validity is hard to obtain because it contradicts variability control. In so-called lab experiments, the comparison between two systems is framed so that variability is controlled and cancels out across subjects and evaluation

conditions. Ecologically valid conditions will ignore those factors and focus on the data and subject distribution to be as representative of the real distribution as possible. For example, if you want to assess how journalists use information collected from the web as background for writing an article, you would want to control how much of the web they can access, and (for replicability and to get IID samples) provide them with the same set of inputs. This contradicts the ecological constraint as it does not represent how they work in real life (where they have access to the whole of Internet). Addressing the other challenges tends to reduce ecological validity as we will see.

The choice of the evaluation metric is also challenging. In a professional context, evaluation metrics have been defined, such as the return on investment (ROI) which is very abstract and requires to invest something and measure how much money can be gathered from customers in response to that investment. The problem is that it is hard to guarantee independence in that context since the investment has an impact on the market and the market is a whole, and it takes a lot of time and resources to run such kinds of experiments. Human resources have also defined metrics for evaluating the work of employees, but they are often computed against an inadequate time span (for example number of successful projects yearly), and they are too subjective to allow for replication. As an evaluation metric, there is often a tradeoff between the time required to perform the task and the quality of the performance. One way to address that problem is to impose a time limit to pressure subjects to perform the task, and then only evaluate quality. Putting pressure on subjects will increase the difference between conditions, but it decreases ecological validity (unless the time pressure is realistic) and it is hard to enforce for tasks where you typically don't have strong time constraints. In addition to task evaluation, it is common to pass the subjects through self-assessment subjective questionnaires (did you like system A better than system B? do you think you did better on task A or B? etc.)

It is important to define a baseline condition that is most similar to the target condition so that it does not impact how subjects perceive the task. Ideally, both conditions should be run under the same user interface (and similar user experience) so that subjects cannot identify the target and perform better just because they figured they should (similar to the placebo effect). In addition, if the user interfaces between the baseline and the target are different, the evaluation will inevitably focus on which UI is better, not which underlying technology is better. There is also an ecological validity problem with technology which provides novel features. The baseline condition should be as similar as possible to that of existing tools to support the task, but exposing novel features provided by the target technology will drive the setup away from ecological validity.

Variability control is hard as in all experimental sciences. It is often associated with psychological biases which shall affect experiment design[4]. The first problem is the effect of order. If a subject passes the test under different conditions, then the fact that a better-performing system is first presented will have an impact on the system presented after it. If subjects are not proficient with the system before using it, there will be a learning effect after each trial, which will make subjects perform better and better. These two effects can be canceled out by adopting a per-subject Latin square design in which the set of all permutations of condition / trial number are tested. With many conditions, Latin squares can get very expensive to run. Subject proficiency in the task is also a problem as it may incur variability in how the subject perform independent of the system. Factors to explain such variability are often collected as cognitive proficiency and attention tests run before the experiment. Variability can also be found in the difficulty of the trials. Let's assume subjects have to process documents, for better ecological validity the trials should be designed to represent the natural distribution of document length and topic difficulty, but this will incur additional variability which will impact the results.

The last challenge is that the variability incurred by the system on subject performance should be large enough to measure a significant difference. And if that system comes early in the pipeline, or the difference between the baseline and target conditions is too small, then the outcome of the experiment will be difficult to interpret. It is therefore very important to calibrate the setup so that differences have a chance of being significant (by choosing a task linked to the system output, by comparing

[4]There are many cognitive biases that affect a range of human capabilities. It is hard to imagine designing systems that are not affected by, or account for all those biases (Benson 2017).

systems with a range of performance…). In addition to impacting ecological validity, this selection of the evaluation setup is more problematic as it is driven by the significance of the differences and therefore biased towards showing that significance exists which would invalidate the breadth of the finding.

So, under the light of those challenges, the perfect experimental conditions is something that should be aimed for, but that is hard to achieve in practice. We will now see two experiments that I have been involved with which exemplify end-user evaluations, their challenges and what to expect from them.

5.3.2 Lessons Learned from two Experiments

In this subsection, I will describe two projects in which we tried to perform an end-user evaluation of NLP technologies. The first one was developed during the Johns Hopkins's summer workshop in 2012 in order to create better measures of automatic speech transcription quality. The second one was explored during the FP7 SENSEI project in order to evaluate conversation summarization technologies.

Johns Hopkins Workshop The objective of the Complementary Evaluation Measures for Speech Transcription (CEMST) team at the JHU workshop organized by the CSLP group in summer 2012 was to come up with improvements over word error rate (WER) as an evaluation metric for automatic speech transcription. At the time, the main disadvantage of WER as an evaluation metric was that it would treat equally "small errors" that did not impact that much how understandable the transcript was and "big errors" that would harm how one could understand the result. While there had been many studies of the impact of ASR on subsequent tasks, such as (Jannet et al. 2015; Zhou et al. 2007), studies of the impact of ASR on end-user performance were few, and generally not aimed at ecological validity.

The CEMST approach was an extension of a study which had set up an extrinsic evaluation of speech summarization technologies under different ASR conditions (Murray et al. 2009). The idea was to pitch several ASR systems, each tuned under different settings in order to yield a range of WER, as input of a meeting summarization system on the AMI dataset. That dataset contains meeting recordings of people role-playing the design of a new product (a remote control) with simulated market feedback. Students then played the role of a newly hired manager which was tasked with listing the design decisions adopted during the meetings by using a meeting summarization system fed with ASR output. Users were evaluated according to their capacity to recover a gold-standard list of decisions and provide evidence from the meetings to support that list. The complete setup is presented in (Favre et al. 2013b†).

The outcome of that experiment was that WER and its variants were not good predictors of user performance but that a system could be trained to combine evidence from a range of features involving the transcript, but also designed to try to factor out user and task effects. In a leave-one-ASR-out experiment, this system proved to be better than WER at predicting user performance but far from perfect (an F-score of about 40% whereas WER was under 5%).

The complexity of running such experiment, the potentially small impact of the ASR transcript on user performance and the improvable ecological validity (non-natural meetings from the AMI corpus and students playing the role of a manager) show the importance of well-preparing such studies.

SENSEI One of the corner stone propositions of the SENSEI project was to run an end-user evaluation of the proposed technology using real-world professionals performing a task that they were used to do as part of their daily work, thereby achieving better ecological validity than the JHU workshop project.

The SENSEI project studied methods for textual and spoken conversation analysis using a structuration chain spanning syntax, semantic, discourse and para-semantic parsing. Here, by para-semantic we mean features of a conversation which do not fit in classic semantic models such as FrameNet, like sentiment and emotion analysis, leveraging user-provided annotations such as "likes", user-provided

moderation, and authorship analysis. The goal was then to leverage the predicted conversation structure in order to summarize them under the angle of one of two use cases: speech analytics in a call center, and user comments of online news (the speech and social media use cases).

For the speech use case, end-user evaluation involved Quality Assurance professionals from a call center whose job is to assess the behavior of agents by listening to conversations in the background and filling evaluation forms with criteria that range from waiting time and agent politeness to customer satisfaction and problem resolution status. The proposed end-user evaluation was to confront a baseline condition where QA professionals had access to the audio and ASR transcript of a conversation and a simple search engine, as provided by standard analytics software, and a so-called SENSEI condition which included features from the semantic and para-semantic pipeline, an advanced search engine and summaries generated for the purpose of the task. Under those two conditions, the subjects had to perform agent behavior assessment and complex information gathering tasks from call-center conversation recordings from the RATP-DECODA corpus (public transportation hotline) and the LUNA corpus (computer help desk at a company).

The first challenge in this setup was that the QA professionals employed were not used to having analytics software available in their daily work, so they had to learn how to use the technology. The technological novelty of the SENSEI conditions also meant that the user experience would play a large part in the experiment, and that subjects could easily recognize in which conditions they were being evaluated, imposing a bias towards the target condition. Finally, the industrial context of outsourced call centers precluded the use of real customer data, which imposed the use of pre-existing corpora such as DECODA and LUNA. It turned out to be a problem because the kind of call centers they were recorded in did not involve a large turnover with agents of variable seniority, and therefore QA professionals found it difficult to evaluate the conversations under the typical criteria they were used to work with. Even though we had them annotate a large set of conversations with evaluation forms, the relatively low inter-annotator agreement resulted in difficulties to build prediction systems for the particular criteria they were evaluating.

The social-media use case was faced with a different challenge. It involved journalists who were tasked to write a summary of the reader comments originated from a news article. Leveraging online readers' comments is a relatively novel activity in the news industry who tries to foster a two-way connection between journalists and their readership. So the main challenge was to build a manual comment summarization methodology in which subtasks could be performed by a system (such as comment clustering, cluster labeling, and summary generation). Such a methodology was necessary in order to rationalize the steps taken by journalists to perform the task and ensure minimum agreement in their understanding of the task. For this end-user evaluation, the main challenge turned out to be the fact that journalists were not used to having NLP technology available for performing their daily work, and therefore had to carry the experiment in a setup that was different from their usual setup. Another limitation of that experiment was that we were not able to recruit actual journalists, but had to fallback to journalism students, due to factors which might have been linked to the novelty of the project and the busy schedule of the profession.

To conclude on the two experiments that I was involved in, the SENSEI project addressed ecological validity in a much more sensible way than the JHU workshop, but it exposed the differential between the baseline and target conditions (creating a bias towards the latter), and had trouble with the technological gap between the professional reality and the state of the art of research-built systems. Another strong challenge was to build the incentive for professionals to take part in end-user experiments.

5.3.3 Impact on machine learning

If ecologically valid end-user evaluation can tell us more than traditional evaluation to which extent a system is able to concretely solve real-world problems, it remains very complex and tedious to run. Complexity comes from the fact that many biases may have an impact on subject performance, due to human factors which are not well appreciated in the NLP community. In particular, creating a good

experimentation setup is difficult and requires multiple iterations before it is able to show any kind of effect. Building collaborations with other communities such as Human Computer Interaction (HCI) would probably help in that regard. The approach is also much more demanding than corpus-based evaluation as it requires collecting novel data every time a system parameter is changed.

In the framework of machine learning, subject performance could be minimized as a loss. However, several problems have to be addressed. The first problem is to untangle the effect of the system and the effect of the subject, experimental setup or the task. One way to obtain better supervision signal may be to work on the residual resulting from a prediction of subject performance from known factors (akin to multiple regressions). If these effects are not untangled, the machine learning model will have to use modeling power to account for them. The second problem is the low reliability of the loss function. Data collected from human experiments typically shows variability beyond the factors accounted for, leading to noise in the data series that would be used as a supervision loss. The machine learning community has long worked on the problem of noisy supervision, and results from the area of semi-supervision or reinforcement learning might be interesting (Gao et al. 2018). Another problem is the generality of the experimental setup. Ecological validity implies the choice of a range of applications of the NLP system being evaluated in a reasonable scenario. This results in other applications being ignored, and the applicability of the learned system as a module of a more general system being reduced. Introducing a smooth transition between corpus-based evaluation and ecological evaluation, through pretraining for example, could be a way of reducing this effect. The last problem we discuss is the problem of the cost of evaluating the loss function. Machine learning typically performs extensive number of evaluations of the loss function, in order to progressively improve models. However, for ecological evaluation, the loss function involves humans performing a task and is therefore quite expensive to perform. Several options are possible for addressing the cost of evaluating the loss function. Black-box optimization allows to perform function optimization with very few samples by, for example, assuming a locally quadratic surrogate (Berghen et al. 2005). Another way of tackling the issue would be to integrate the cost of evaluation in the loss function and jointly optimize for good system performance and reduced evaluation cost. It would be, however, rather difficult to balance the two objectives. Finally, one could also use a proxy loss updated infrequently from the real loss, as we proposed in (Favre et al. 2013b†).

In (2013b†), we built a classifier that predicts user performance from a range of features extracted from the task, the users, and the ASR system being evaluated. This classifier was based on boosted decision stumps for their robustness to irrelevant features, an expected phenomenon when generating as many features as possible. As already stated, the experiment showed that intrinsic evaluation metrics alone were poor predictors of user performance. It is a step in the right direction, but inferring system performance outside of the range of evaluated systems is difficult as is any extrapolation problem, and the system sample size was probably too small to draw any conclusion. A lot of work is required in this area in order to find a definitive approach.

5.4 Conclusion

Evaluation is at the heart of the scientific methodology for measuring progress in NLP. The traditional approach consists in creating annotated corpora, specifying metrics and computing the performance of proposed systems given those metrics. In this chapter, we have seen that there are challenges with this methodology. The first is that relying on the same corpora, even if it ensures reproduction of scientific results (same data and software lead to same results), it does not ensure replication (collecting new data in the same conditions leads to the same results). Shared tasks associate third-party judgments and tests on unseen data, and even though they do not aim at reproduction, they provide a good framework for system benchmarking. Nevertheless, they focus on engineering[5] over addressing research questions, and the absolute results they generate depend on high participation rates, at a cost of diversity. By over-relying on the community, shared tasks also remove one of the benefits they were created for:

[5]I am not aware of any study showing the effect of over-engineering a proposed approach versus the baseline it is confronted to. It would be interesting to research that area.

the evaluators are not truly third parties anymore but members of the community which have built sometimes conflicting interests over time.

Secondly, even if exemplary experimental conditions can be obtained, the value of results highly depends on the task being benchmarked. We advocate for ecologically valid measurement of the impact of NLP systems on user performance, on tasks that matter to them. However, this setting is hard to exploit in the context of machine-learning and it suffers from hard-to-obtain generality. How to untangle the effect of presentation (user experience) and the effect of technology? Users performance is affected by design issues, which might explain more variability than the technology being evaluated. How can we evaluate systems that allow to do tasks that subjects did not do before? For example, measuring the impact of Internet searches on journalists work at the times when no Internet search was available would have been hard because the technology being evaluated bring new uses which were not possible before.

The industry might be better armed to tackle the evaluation problem, given for instance the pervasively used A/B testing or multi-armed bandit approach when developing web applications. In a world of continuous integration, randomly drawn sets of users are presented with different versions of an approach (design, features, underlying algorithms), and the one selected for large deployment. Under valid evaluation metrics, such approaches are known to improve the technological outcome, but how can we follow them in the NLP community if we have neither the platforms nor the user base?

As for devising a good methodology in the context of machine learning, back-propagating user success to algorithms seems to be an interesting direction, and a call for even tighter integration with the underlying statistical frameworks and assumptions. Can we imagine including significance tests in the loss function?

PhD student work relevant to this chapter

- **Thibault Magallon** participated to the ICDAR 2017 post-OCR error detection and correction shared task (Magallon et al. 2018a[†]). Most of my students have worked with shared-task data, experiencing the traction of engineering challenges, and the difficulty of replicating published work.
- **Jérémy Trione** contributed to the extrinsic evaluation framework in the SENSEI project, in particular he created the system for automatic summarization of french conversations (Trione et al. 2016a).
- **Simone Fuscone** works on the characterization of conversational interpersonal dynamics. He has replicating past speaker convergence work and addressing the gaps left out in published papers on the topic (Fuscone et al. 2018[†]).
- **Sebastien Delecraz** released an annotated corpus of preposition attachments in image captions with alignments between texts and visual data (Delecraz et al. 2018a[†]).

Chapter 6

Conclusion & Prospects

Natural language is pervasive in a society of information and communication. Content on the Internet, be it for work or leisure, mostly consists of unstructured natural linguistic exchanges between humans. Even though video is progressively replacing text as a medium, and may eventually replace it, both are intrinsically made of constructed narratives expressed through natural language. Be it for automation, marketing, governance or art, the need for natural language processing is increasing, especially in an era of renewed interest for artificial intelligence.

This document first summaries my activity in the field of NLP. Then, it develops arguments around the idea that to move forward, the NLP community needs to look at the problems it tackles from a more contextual point of view. These arguments are partly based on my experience and results obtained through my work, and partly based on an analysis of the findings presented in the community.

A first chapter introduces the main concepts of NLP as an empirical science, and its current rooting in statistical modeling and in particular machine learning. This primer explains the virtuous loop which allows progress in the field (which I call TCGASE for task, corpus, guide, annotation, system, evaluation). It then abstracts the typical NLP tasks as a set of five underlying tasks: labeling, segmentation, regression, relation detection and generation, which can be framed as machine learning problems with relative ease. This sets the frame for presenting how machine learning is applied to NLP and how it evolved in the past 15 years, from shallow models to deep models. The latest developments in deep learning, which have often been driven by NLP problems and the NLP community, allow a much more straightforward global learning of the parameters of composed system blocks. Machine learning relies on a cost function evaluating how well the parameters of a system allow it to replicate an expected output. In NLP, evaluation of the success of a system is not always straightforward because multiple outputs may be acceptable, and because judges may not agree on what a good output will look like. This leads to convoluted evaluation functions which are difficult to approximate with derivable cost functions appraised in the context of machine learning.

The next chapter questions the typical assumptions behind NLP models and systems, in part inherited from statistical machine learning, and often necessary for ensuring computational tractability. It assesses whether the typical dependencies assumed at the input, decision and phenomenon levels are the most representative of reality and the most effective for performant systems. The human-performance hypothesis states that systems modeled after human cognition should only require input representations that are necessary for a human to perform the task. This hypothesis leads to subpar performance because systems are much less sensitive than humans to stimuli and because human cognition heavily relies on past experience and is the result of an evolution process which cannot be replicated in controlled lab conditions. We show that for multimodal NLP tasks, contextualizing decisions with input representations that go beyond the conventional monomodal input leads to performance improvements in a set of tasks. The second assumption studied is that of structured predictions which compose a labeling of the input from elemental labels. Models under the Markov assumption are a good example of models that account for a local set of predictions when seeking for a global

optimal structure. These models have dominated a lot of fields in NLP where structured predictions are necessary, but are fading in the context of deep learning, in a movement to trade weak models with strong decoders for strong models with weak decoders, maybe at a constant computational requirement. However, when structure constraints are complex and modeling is hard, such as in automatic summarization, weak models with strong inference might continue to be necessary. The last explored assumption is that of processing chains which sequentially solve tasks addressing different phenomena depending on each other. We show that end-to-end learning of deep models which assume implicit decomposition of the phenomena can be improved by supervising them to recognize all phenomena in a single model, in a multitask fashion.

The third chapter is dedicated to the generalization of NLP findings. It first defines a difficulty scale for generalizing (distributional sample, time frame, genre, topic, language, task) and explores, through a set of examples, how systems can be said to generalize at each of those levels. We first describe how speech is a particular genre for NLP that requires specific processing compared to text, the de facto most common modality for NLP, and show a few directions for better generalizing when crossing the boundary of the genre. At the input level, word embeddings, a technique for creating word representations based on frequent associations, need to be synthesized for low-frequency and unknown words. The community has also shown that building on characters (or parts of the input, which would be acoustics for speech), and contextualizing representations is a good way to obtain generalizable input representations. In the context of deep learning, a well-established technique is to build on characters, and pretrain lower layers of the systems on language modeling tasks on very large unlabeled text corpora. In terms of generalizing across tasks, multitask learning which shares the parameters of systems trained to perform different tasks may lead to universal parameter spaces that are good for many tasks. Two questions arise: how to find an optimal weighting of the tasks, and whether the compromise of learning a general system, leading to lower performance on a particular task, can be alleviated on unseen tasks. Finally, generalizing across languages is explored. This problem, which is of great interest for the community, requires exploiting multilingual invariants that both allow modeling cross-lingual phenomena and modeling the specifics of individual languages.

The last chapter questions the methodology used by the NLP community to foster progress. First, it looks into shared tasks as a way of insuring independent evaluation of system performance. Shared tasks suffer from over focusing on engineering, putting less effort on scientific understanding and more effort on the outcome, often leading to a closer-sighted scientific vision. Shared tasks also put too much light on specific problems, reducing diversity at a constant effort, and the tendency to advocate the organization of shared tasks by project funding agency, which has a positive effect on diversity, tends to lower participation rates, and therefore reduce the scope of results. We promote the reconsideration of the TCGASE loop by evaluating systems not on annotated corpora, but by measuring the performance of users of those systems in the context of ecological tasks. The experience from two use cases shows that although more representative, such evaluation methodology is costly and difficult to run in a community not used to work with human subjects.

To conclude, natural language processing is a field with great societal expectations. It fuels the field of artificial intelligence, and might be key to developing artificial though processes. It also fuels a range of sciences, be it linguistics which use it to develop theories grounded in corpora, or psychology and cognitive sciences aiming at developing a better understanding of how the brain manipulates language, and it also bears a great applicative potential in fields such as medicine, law, history which would benefit from better modeling of documents and interactions. And of course, in the industry, applications will continue to thrive as language is one of the main forms of communication enjoyed by humans. We have seen in this document that achieving progress in natural language processing will require a rigorous methodological framework that encompasses the study of linguistic and applicative invariants and the generalization properties of systems. It will only be possible with a collaboration between the numerous fields that breed to form contextual NLP.

Part II

Summary of activity

Chapter 7

Curriculum Vitæ

Identity	Benoit Favre
Date of birth	1980-07-14
Job	lecturer / associate professor (MCF Section CNU 27)
University	Aix-Marseille Université
Lab	Laboratoire d'Informatique et Systèmes (LIS), data science group
Doctoral school	École Doctorale en Mathématiques et Informatique de Marseille - ED184
Website	https://pageperso.lis-lab.fr/benoit.favre

7.1 Summary

- Associate professor at Aix-Marseille University (France) since 2010
- Research
  - Head of the data science group at LIS, member of the scientific piloting committee of LIS
  - PEDR 2016-2020
  - 112 scientific publications
  - Projects:
    - coordinator: 1 FP7 EU
    - participant: 3 DARPA, 9 ANR, 1 AMIDEX, 1 ERC
  - PhD students: 8 (4 graduated)
- Teaching
  - Head of the Artificial Intelligence and Machine learning MSc track at AMU
  - $\geq 1,300$h experience in computer science in BSc and MSc degrees
- Keywords: automatic conversation understanding, automatic summarization, natural language processing, deep learning, representation learning, machine learning, evaluation

7.2 Scientific activity

7.2.1 Awards

- PEDR scientific excellence 2016-2020.
- Best paper at TALN 2011 (Roux et al. 2011†)
- Best paper at IWANN 2019 (Delecraz et al. 2019†)

7.2.2 PhD Students

I advised or am advising 8 PhD students (4 graduated):

- Manon Scholivet, 2017+, "Language-independent natural language processing", funded by doctoral school ED184 (33% with C. Ramisch & A. Nasr) – Leveraging language structure representations and language-independent input representations to process never-seen languages. Application to parsing.
- Simone Fuscone, 2017+, "Linguistic Underpinnings of Conversational Interpersonal Dynamics", DOC2AMU (50% with L. Prevot) - We explore the evolution of convergence and divergence phenomena in conversational context, in order to discover interpersonal behavior profiles.
- Jeremy Auguste, 2016+, "Conversation representations", funded by ANR Datcha (50% with F. Bechet) - This thesis aims at devising compact and relevant conversation representations from raw and analyzed goal-oriented dialogs.
- Thibault Magallon, 2015+, "Global features for optical character recognition", CIFRE funded with Numericompta (50% with F. Béchet) – The objective of this thesis is to create global inference methods for analysing documents with optical character recognition.
- Sébastien Delecraz, 2015-2018, "Multimodal language understanding", AMU-DGA funding (50% with F. Béchet) – This thesis explores multimodal natural language processing, and document understanding from multimodal sources.
- Olivier Michalon, 2013-2017, "Semantic frame parsing of French", funded on ANR project ASFALDA (50% with A. Nasr) – This thesis explores Framenet parsing with global inference and exogenous representations.
- Jérémie Tafforeau, 2013-2017, "Multitask inference for natural language processing", funded by FP7 project SENSEI (33% with T. Artières and F. Béchet) – In this thesis, we explore a generic neural network topology for learning multitask sequence predictions and apply it to NLP problems.
- Jérémy Trione, 2013-2017, "Automatic abstractive summarization of conversations", funded on FP7 project SENSEI (50% with F. Béchet) – this thesis proposes abstractive summarization methods based on template generation and filling.

7.2.3 Master Students

I also supervised 6 MSc students:

- Marjorie Armando, 2018, "Interpretable textual inference models"
- Manon Scholivet (co-supervised with C. Ramisch), 2017, "Polylexical expression detection with RNNs"
- Jeremy Auguste (co-supervised with F. Bechet), 2016, "Conversation representations for analytics"
- Sébastien Delecraz (co-supervised with F. Bechet), 2015, "Multimodal understanding of video documents"
- Jérémy Ollivier, 2013, "Paraphrase detection"
- Firas Hmida, 2011, "Language-independent automatic summarization"

7.2.4 Projects

I have been involved in multiple French ANR projects, one EU FP7 project, and 3 DARPA projects:

- DATCHA, ANR, 2015-2019. Automatic analysis of text conversations from customer care centers. Partners: Aix-Marseille Université, IRIT, Orange. In this project, I work on representation learning of gold-oriented conversations, as well as the relationship between semantic and discourse representations.
- PARSEME-FR, ANR / COST PARSEME, 2015-2020. Syntactic parsing and multiword expressions in French. Partners: Aix-Marseille Université (LIF), LIGM, INRIA, LI, LIFO. I was involved in the study of novel machine learning techniques for MWE detection and discovery.
- ADNVideo, A*midex (IDEX from Aix-Marseille Université) for technology transfer towards the industry, 2014-2017. Multimodal recommendation of videos. Partners: Aix-Marseille Université (LIF, LSIS), Kalyzee. I was involved in the transfer of video content analysis modules mainly designed during the PERCOL project.
- SENSEI, European FP7, 2013-2016 Making sense of human-human interactions. Partners: Aix-Marseille Univ., U. Trento, U. Sheffield, U. Essex, Websays, Teleperformance. I was the **local coordinator** for this project. I have contributed to evaluation organization, semantic parsing of conversations, abstractive summarization of call-center conversations.
- ORFEO, ANR, 2012-2016. Creation of a transcribed and annotated corpus of spoken and written French. Partners: Aix-Marseille Univ., LATTICE, MODYCO, ATILF, LORIA, CLLE-ERSS, ICAR. I was involved in automatic sentence segmentation as well as automatic processing of conversations with the Macaon chain.
- ASFALDA, ANR, 2012-2015. Semantic parsing of French. Partners: Aix-Marseille Univ., Alpage, CEA, IRIT, LLF. I was involved in designing automatic inference techniques for semantic parsing with the Framenet framework, as well as in the introduction of exogenous sources in parsers.
- PERCOL, ANR, 2011-2014. Person recognition in videos. Partners: Aix-Marseille Univ., Orange, LIA, LIFL. I designed multiple modules of the video processing chain, and a multimodal fusion system for person recognition.
- DECODA, ANR, 2010-2013. Spoken conversation processing in call centers. Partners: Aix-Marseille Univ., UAPV, RATP, SONEAR. I was involved in building syntactic parsers adapted to spoken language.
- SEQUOIA, ANR, 2010-2013. Syntactic parsing of French. Partners: Aix-Marseille Univ., Alpage, LIA, LaLIC. My contribution consisted in designing CRF-based part-of-speech taggers, as well as graph-based dependency parsers.
- ThunderBOLT, DARPA, 2012-2015. Speech-to-speech translation with clarification dialogs. Parters: SRI International, U. Washington, U. Rochester, U. Columbia. I was involved in ASR and MT error detection, as well as transcript correction after clarification dialogs.
- PORTMEDIA, ANR, 2009-2012. Robustness, portability of semantic parsing in dialog systems. Partners: LIA, LIG, LIUM, LORIA, ELDA. I worked on concept detection and ASR.
- CALO, DARPA, 2003-2008. Cognitive assistant that learns and organizes. Partners: ICSI, SRI International (and others). I contributed automatic summarization techniques.
- NightinGALE, DARPA, 2005-2010. Distillation from multilingual, multigenre speech and text. Partners: ICSI, SRI International, U. Columbia, U. Washington, NYU (and others). I was involved in designing multilingual sentence segmentation and punctuation prediction models.

7.2.5 Shared tasks

I participated to and organized shared tasks (scientific competitions) in the natural language processing community.

- Sentiment analysis: Semeval 2016 (ranked 2/34), DEFT 2015 (1st), TASS 2015 (2nd). With M. Rouvier.

- Multimodal person identification in videos: REPERE 2013-2015. Ranked 1st with the PERCOL consortium.
- Automatic summarization: organizer of Multiling 2011, 2015, participant to TAC 2006, 2007, 2008 (ranked 2/33), 2009 (ranked 1/30)
- Referential expression generation: GREC evaluation in 2009 (ranked 1st) and 2010 (1st).
- Named entity recognition: ESTER and ETAPE evaluations in 2005 and 2012.
- Speech-to-speech translation: DARPA BOLT 2012-2014 with the Thunderbolt team.
- Multimodal, multilingual document retrieval: DARPA GALE 2007-2009, with the Nightingale team.

7.3 Teaching

I am the head of the **artificial intelligence and machine learning** track of the computer science Master at the CS department of Aix-Marseille University. I have led the creation of this track in 2017-2018.

In 2017, I was awarded a full teaching service reduction by CNRS (delegation). In 2015, I was awarded a partial teaching service reduction of 69h by the department (service modulation).

I have performed a full service of teaching at Aix-Marseille University since 2010 (192h per year + overtime). Main classes:

- Deep learning for NLP – 2016+ (35h/y), MSc level (M2), RNNs, Language models, MT, representation learning, applications.
- Web programming – 2012-2015 (30h/y), BSc level (L2), Javascript, node.js, html5.
- Languages theory – 2011-2015 (50h/y), BSc level (L2), Finite state automata, regular languages.
- Data structures – 2015 (20h/y), BSc level (L3 pro), Algorithms, complexity, basic data structures.
- Programming methodology – 2011-2014 (38h/y), MSc level (M2), UML, design patterns, continuous integration.
- C programming – 2016 (128h), 2010-2015 (50h/y), BSc level (L1), Basics of programming, data structures, pointers, sorting algorithms, linear data structures.
- Natural language processing – 2010-2014 (35h/y), MSc level (M1), Statistical approaches, methodology, evaluation, machine learning, sequence predictions, machine translation, automatic summarization.
- New applications of NLP – 2013-2014 (35h/y), MSc level (M2), machine translation, sentiment analysis, parsing, summarization.
- Computing tools 2010-2011, 2014 (30h/y), BSc level (L1), OpenOffice, latex, shell, certification training (C2i).
- XML – 2010-2011 (33h/y), MSc level (M1), DTD, schemas, stylesheets, Java api.
- Digital sciences (ISN) – 2012-2015 (12h/y), secondary teachers (math, physics, CS), language theory, python.
- Student projects – since 2010, MSc level (M1/M2), multiple projects per year on a variety of topics.

In 2017, I taught a Deep learning for NLP class at IIT Mizoram at Aizwal in India (GIAN program). In 2018, I have organized two sessions of that class with CNRS Formation towards the industry[1].

[1] https://cnrsformation.cnrs.fr/stage-18289-Deep-learning-pour-le-traitement-automatique-des-langues.html

In 2017, I obtained a AMU FIP Grant for the GPU-Training program which supported the acquisition of GPU servers for organizing a range of classes under the computer science MSc (Deep learning, GPU programming, distributed computing...). The budget of the project was about 60k euros (co-funded by the CS dept, the Economy and management dept, ILCB and the FIP).

I am a member of the teaching committee of the Institute for Language, Communication and the Brain (ILCB).

7.4 Administration and scientific animation

Since 2018, I am the **head of the data science group** at LIS, gathering 5 research teams (44 permanent researchers). Duties include defining and implementing a scientific policy, organizing scientific animation for the members of the group, managing the budget of the group, interfacing with the head of the lab, and involvement in the governing body of the lab. Since 2017, prior to the creation of LIS (which results from the fusion of LIF and LSIS), I acted as head-elect to define and implement the governance of the new lab.

I am part of the machine learning and deep learning "question transverse" group at the Institute for Language and the Brain (ILCB), a convergence institute under Amidex. I have been working on multiple projects relevant to ILCB, and have been repeatedly presenting my work in that context. I obtained two grants from ILCB (one in collaboration with A. Rey from LPC and one with multiple members of LIS).

I contributed to building a computation cluster at LIF, including getting hardware grants, defining and managing a use policy, becoming a NVIDIA teaching and research center (which granted 2 GPUs). Since 2018, I am responsible for the "commission cluster" at LIS, in charge of defining a strategy for building a viable computation platform for the members of the lab, including relations with institutional efforts for computation (AMU mesocentre and regional centers).

In 2018, I contributed to the emergence of a CNRS GDR (Groupement de Recherche) on natural language processing. I was in charge with P. Sebillot (IRISA, Rennes) of writing a report on the state of the art and future trends of intermodal and multimodal natural language processing.

I was part of the board of AFCP (French speech society) 2014-2016, in charge of student grants, and since 2017 I have been part of the board of ATALA (French NLP society).

I co-organize a deep learning reading group at Aix-Marseille University since 2017 with T. Artieres.

In 2017, I was a member of the "commission web" of LIS tasked with defining the web presence of the merged lab.

I was responsible for organizing the seminar of the NLP group at LIF in 2015-2016.

7.4.1 Conference committees

I was part of the organization for the following conferences:

- Summer school of the Institute for Language, Communication and the Brain (ILCB), Marseille (France). 2018.
- SemDial 2018, Aix (France). Area chair.
- TALN 2018, Rennes (France). Area chair.
- TALN 2014, Marseille. Local organizer and publication chair.
- SLAM 2013 (Workshop Speech, Language and Audio in Multimedia), Marseille. Local organizer and publicity chair.
- SLT 2010, Berkeley. Publicity chair.

7.4.2 Reviews

I regularly review papers for conferences and journals:

- Journals: IEEE transactions on speech and language processing, Speech Communication, IEEE Signal Processing Letters (4 per year)
- International Conferences: ACL, EMNLP, COLING, NAACL, ICASSP, Interspeech, SLT, ASRU, NIPS (5 per year)
- National conferences: TALN, JEP, ReciTAL, CORIA (3 per year)

7.4.3 Expert

I have been an expert for ANR since 2013 (5 projects evaluated). I was an expert for the EU ERC program (2015).

I was an expert for the following associate professor hiring committees in French universities (computer science, CNU 27):

- Paris Sud, 2018
- Ecole Centrale Marseille, 2017
- Orléans, 2015, 2016
- AMU, 2012, 2013
- Paris 1, 2012

I participated to the following thesis defense committees:

- Qi Wong (LIS, AMU), 2018
- Sahar Ghannay (LIUM, Le Mans), 2017
- Mohammed Bouaziz (LIA, Avignon), 2017
- Raheel Kader (IRISA, Lannion), 2017
- Jeremy Trione (LIS, AMU), 2017
- Jeremy Tafforeau (LIS, AMU), 2017
- Olivier Michallon (LIS, AMU), 2017
- Anca Simon (IRISA, Rennes), 2015
- Pierre Gotab (LIA, Avignon), 2012

I was a member of the thesis steering committees (comité de suivi de thèse) of:

- Julien Dejasmin (LIS, AMU, 2019-)
- Luc Giffon (LIS, AMU, 2017-)
- Antoine Caubrière (LIUM, Le Mans, 2017-)
- Ozan Caglayan (LIUM, Le Mans, 2015-2019)
- Sahar Gannay (LIUM, Le Mans, 2015-2017)
- Xavier Bost (LIA, Avignon, 2015-2017)

I was an external expert for the doctoral grant committee at UAPV (Avignon) in 2013.

7.5 Dissemination

7.5.1 Invitations & scientific talks

I was invited as a research scholar at Queensland University of Technology for 9 months in 2018-2019.

I gave a talk at the PARSEME ANR project on deep learning for polylexical expression detection in 2018.

I gave a talk on the current state of deep learning and its link with the origins of language in the brain at the ILCB workshops in 2015 and 2017.

I have joined a pan-European group on multilingual summarization in order to federate the community and organize a COST network (2015-2017).

I was invited to the Johns Hopkins University workshop for 6 weeks in 2012 as a mentor on the CEMST project on speech transcription evaluation measures. I supervised students and postdocs on the topic.

I was an invited speaker at an NSF workshop on the future of automatic summarization in 2012.

I repeatedly presented my scientific work at LIS and at the scientific days of LIS.

7.5.2 Dissemination towards the general public

I was invited to a panel on the ethics of artificial intelligence by LICA at EMC in Marseille in 2017.

I participated to meetups on machine learning in the local area of Marseille in 2016-2017.

I have collaborated with artists from the Pulso company since 2015 on the relationship between humans and machines. I supplied the technical framework of the Homeostatis dance show in 2016-2018, participated to the Ergonomics performance at the CDA of Enghen-les-bains in 2017, co-authored the IAgotchi installation at Experimenta in 2018. I was also invited to a workshop on Art-Science in 2017 on the topic of artificial intelligence.

I participated to the European night of researchers in September 2017 and 2018 for the city of Marseille.

7.5.3 Software

I commit to delivering as often as possible open-source implementations and data related to my scientific work. I created or contributed to a number of software libraries and applications, such as:

- ICSIBoost: open-source implementation of Boostexter (Adaboost based classifier, used by many researchers in the NLP community)
- Macaon: NLP processing chain (tagging, parsing...)
- ADNVideo: multimodal video analysis tools
- ICSISumm: extractive summarizer based on the budgeted set-cover problem (ranked 1st at TAC 2009)

My contributions are available in the following repositories:

- https://github.com/benob
- https://gitlab.lif.univ-mrs.fr/u/benoit.favre (listed as archived projects)
- https://gitlab.lis-lab.fr/u/benoit.favre

7.5.4 Publications

I am the co-author of 112 publications: 4 book chapters, 9 peer-reviewed international journal articles (+1 submitted), 69 peer-reviewed international conference articles, 16 peer-reviewed French conference articles, 14 other publications including theses and reports. H-Index of 22.

Chapter 9 lists my publications according to those categories.

Chapter 8

Detail of PhD students' work

8.1 Olivier Michalon

Statistical models for predicting semantic frames 2013–2017, funded on ANR project ASFALDA (50% with A. Nasr)

Summary Semantic parsing consists in predicting actionable representations of the sense of an utterance based on the words it contains. The many semantic models that have been proposed over the years make different compromises between deepness of the representation, coverage and applicability, robustness that can be expected from a system implementing it, and availability of annotated training data. The FrameNet model assumes a sweet spot in that landscape with a high coverage, relatively surfacic representations, and availability of training corpora, in particular thanks to the ASFALDA project which has the objective of building a large scale FrameNet corpus for French. A frame describes a semantic situation that can be expressed through a text. It is evoked by a trigger word and linked to a number of arguments subject to constraints. There can be many frames evoked in an utterance, and frames are organized in an ontology. Even though a number of systems have been proposed for frame parsing, their robustness regarding certain design decisions remains unknown. This thesis explores the link between syntax and semantics through the use of syntactic paths and generalization of syntactic paths as features for semantic parsing. It also compares a number of architectures for implementing task constraints in inference, such as context-free semantic grammars, integer linear programming inference and neural networks. The last contribution is a study of how exogenous data, such as word embeddings trained on a very large raw corpus, impact semantic parsing performance, in particular in terms of training parameters such as window size, input representation and inference mechanism.

Status Graduated on October 4, 2017.

Co-publications

Michalon, Olivier, Corentin Ribeyre, Marie Candito, and Alexis Nasr. 2016. "Deeper syntax for better semantic parsing". In *Coling 2016 - 26th International Conference on Computational Linguistics*. Osaka, Japan.

8.2 Jérémy Trione

Automatic abstractive and extractive summarization of speech conversations from call centers 2013–2017, funded on FP7 project SENSEI (50% with F. Béchet).

Summary Call centers process millions of calls every day to provide information, booking or customer care services. Understanding what happens during the conversations between agents and customers has become a major issue for businesses that use call centers, and great efforts are fostered towards better speech analytics in that context. While state of the art methods propose keyword search on automatic transcripts of what was said in the conversations and prediction of classes of emotions and behaviors of the participants, those techniques do not allow to get a full understanding of what occurred in a conversation. This thesis looks as automatic summarization as a novel way of processing conversational data: the generation of a short textual representation of what was said, and how the participants behaved. While many techniques for extractive summarization had been proposed for speech conversation, they do not respect the style of the expected summaries (well-built narrative vs spontaneous speech transcript) and therefore perform poorly. The main innovation developed in this thesis is an abstractive summarization technique which can create narratives that respect the expected style. The approach relies on a corpus of summaries as training data which have been generalized with concepts to allow them to represent generic situations. Concepts can then be detected in new conversations, and associated with templates automatically extracted from the training summaries. In a sense, this approach shifts the extractive paradigm to use generalized summaries instead of direct transcripts. Experiments show that it outperforms traditional extraction and baseline abstractive techniques. A system based on the approach was also evaluated in the context of the SENSEI project in an extrinsic setup where call-center operators use the generated summaries, showing their benefit in an ecological manner.

Status Graduated on October 5, 2017.

Co-publications

Favre, Benoit, Evgeny Stepanov, Jérémy Trione, Frédéric Béchet, and Giuseppe Riccardi. 2015. "Call Centre Conversation Summarization: A Pilot Task at Multiling 2015". In *Sigdial*.

Stepanov, E., B. Favre, F. Alam, S. Chowdhury, K. Singla, J. Trione, F. Bechet, and G. Riccardi. 2015. "Automatic Summarization of Call-Centre Conversations". In *IEEE ASRU Demo*.

Trione, Jeremy, Frederic Bechet, Benoit Favre, and Alexis Nasr. 2015. "Rapid FrameNet annotation of spoken conversation transcripts". In *Joint ACL-ISO Workshop on Interoperable Semantic Annotation*.

Trione, Jérémy, Benoit Favre, and Frederic Bechet. 2016a. "Détection de concepts pertinents pour le résumé automatique de conversations par recombinaison de patrons". In *Actes de la conférence TALN 2016*. Paris, France: ATALA.

Trione, Jérémy, Benoit Favre, and Frédéric Béchet. 2016b. "Beyond utterance extraction: summary recombination for speech summarization". In *Interspeech, San Francisco (USA)*.

8.3 Jérémie Tafforeau

Multitask inference for natural language processing 2013–2017, funded by FP7 project SENSEI (33% with T. Artières and F. Béchet)

Summary Natural language processing is often considered as a sequence of tasks which add layers of more and more abstract representations: tokenization, part-of-speech tagging, syntactic parsing, semantic parsing… Delivering those tasks in sequence often leads to suboptimal results for two reasons: the propagation of errors from one level to the next and the fact that the information required to disambiguate decisions at a given level is often only available after processing the next levels. Classical natural language processing models in the processing pipeline are also affected by the quality of the feature engineered, as well as domain adaptation. End-to-end models have been proposed to directly

generate labels at the highest level, for instance semantic parsing, directly from words. However, those approaches tend to perform lower than the pipeline approach since they do not have explicit supervision on the intermediary representations, and are not fed with labeled enough data to learn from scratch the involved linguistic phenomena. This thesis proposes a multitask modeling for end-to-end NLP systems which consists in training the model to generate the intermediary labels from the same shared recurrent state, and therefore learn a linguistically informed representation even for the higher-level task. The approach is shown to be more robust to automatic speech recognition phenomena such as word errors and disfluencies. Another contribution lies in the adaptation of input representations in order to generate surrogate word embeddings for unseen words that may occur in novel domains.

Status Graduated on November 20, 2017

Co-publications

Tafforeau, Jeremie, Thierry Artieres, Benoit Favre, and Frederic Bechet. 2015a. "Adapting lexical representation and OOV handling from written to spoken language with word embedding". In *Interspeech*.

Tafforeau, Jeremie, Frederic Bechet, Thierry Artiere, and Benoit Favre. 2016. "Joint syntactic and semantic analysis with a multitask Deep Learning Framework for Spoken Language Understanding". In *Interspeech, San Francisco (USA)*.

Tafforeau, Jeremie, Frederic Bechet, Benoit Favre, and Thierry Artieres. 2015b. "Lexical embedding adaptation for open-domain spoken language understanding". In *NIPS Workshop on Spoken Language Understanding (SLUNIPS)*.

8.4 Sébastien Delecraz

Multimodal language understanding 2015+, AMU-DGA funding (50% with F. Béchet)

Summary While the natural language processing community has mainly focused on textual language, natural language is by definition multimodal, because humans use verbal as well as non-verbal cues in communication, and because our understanding of the real world comes from a multisensorial experience. While there have been many attempts at modeling phenomena which involve another modality such as speech or sign language, there is no definitive approach to multimodal language processing. The main challenges of multimodal language processing are as follows: asynchronicity between phenomena in multiple modalities, difference of annotation, modeling and detection difficulty across modalities, annotator availability in disjoint corpora, lack of theoretical model on multimodal phenomena… This thesis makes two main contributions in that regard. First, a multimodal embedding modeling and fusion approach is proposed and applied to the task of speaker role prediction in video shows using audio, image and transcript-based representations pre-trained on external tasks, fine-tuned on the target task and finally merged to form multimodal representations. The second contribution consists in an approach for supporting syntactic parsing, a task historically developed in a unimodal context, with non-textual modalities. Prepositional attachments, one of the most difficult problems and most common source of errors for syntactic parsers, can be ambiguous for semantic reasons. The proposition is to use an image depicting the text being analyzed in order to resolve ambiguity, and in particular account for the relative position of objects mentioned in the text and visible in the image. Experiments with a system which detects objects in both images and corresponding texts, aligns them to extract features for a preposition attachment error correction module show that when semantic categories of words are not available, a significant portion of attachments can be resolved.

Status Graduated on December 10, 2018

Co-publications

Delecraz, Sebastien, Leonor Becerra-Bonache, Benoit Favre, Alexis Nasr, and Frederic Bechet. 2018a. "Correction automatique d'attachements prépositionnels par utilisation de traits visuels." In *TALN*.

Delecraz, Sebastien, Leonor Becerra-Bonache, Alexis Nasr, Frederic Bechet, and Benoit Favre. 2019. "Visual Disambiguation of Prepositional Phrase Attachments: Multimodal Machine Learning for Syntactic Analysis Correction". In *International Work-Conference on Artificial Neural Networks*, 632–643. Springer.

Delecraz, Sebastien, Frederic Bechet, Benoit Favre, and Mickael Rouvier. 2016. "Fusion d'espaces de représentations multimodaux pour la reconnaissance du rôle du locuteur dans des documents télévisuels". In *Actes de la conférence JEP 2016*. Paris, France: AFCP.

Delecraz, Sebastien, Alexis Nasr, Frederic Bechet, and Benoit Favre. 2018b. "Adding Syntactic Annotations to Flickr30k Entities Corpus for Multimodal Ambiguous Prepositional-Phrase Attachment Resolution". In *LREC*.

— . 2017. "Correcting prepositional phrase attachments using multimodal corpora". In *Proceedings of the 15th International Conference on Parsing Technologies*, 72–77.

Rouvier, Mickael, Sebastien Delecraz, Benoit Favre, Meriem Bendris, and Frederic Bechet. 2015b. "Multimodal Embedding Fusion for Robust Speaker Role Recognition in Video Broadcast". In *IEEE ASRU*.

8.5 Thibault Magallon

Contextual inference for optical character recognition 2015+, CIFRE with Numericompta (50% with F. Béchet)

Summary Optical character recognition (OCR) consists in transforming images containing printed or handwritten text into a digitized form of that text as a sequence of character codes. The task consists in the following steps: global layout understanding which corresponds to detecting text zones, images or tables, followed by line and character detection which locate them from pixels, and finally character recognition which aims at identifying a character given the pixels in the area where it is supposed to be. It is a very relevant task for processing scanned historical documents, and a range of documents which still live a paper trail, such as invoices, receipts, etc. The main objective of this thesis is to improve the quality of OCR output by accounting for the global context in which characters are printed. For example, in an invoice, the recognized numbers of individual items must sum up as the total which is associated to that category. The global context that can be accounted for is mostly constraints dependent on the document type, linguistic constraints such as how words follow each other, and co-variates between linguistic information and global document features such as its color, layout and elements it contains. This thesis outlines two main contributions: the first contribution is the integration of global image representations along with automatic transcripts in a document classification task, leveraging task-specific constraints in order to improve the understanding of the document content, and the second contribution is a strategy for detecting errors in OCR output based on linguistic and visual models, as well as proposing corrections to these errors.

Status As of 2018, Thibault is writing his thesis document and should graduate before the end of the year.

Co-publications

Magallon, Thibault, Frederic Bechet, and Benoit Favre. 2018a. "Détection d'erreurs dans des transcriptions OCR de documents historiques par réseaux de neurones récurrents multi-niveau." In *TALN*.

— . 2018b. “Fusion multimodale image/texte par réseaux de neurones profonds pour la classification de documents imprimés.” In *CORIA*.

8.6 Jeremy Auguste

Learning conversation representations 2016+, funded by ANR Datcha (50% with F. Béchet)

Summary Contact centers are a tool used by businesses to reach out to their customers and offer services such as customer care and product support. Interactions in that framework take the form of human-human conversations over the phone or over textual discussion channels. Many conversations can be collected this way, but it is difficult to mine their content to improve the service, or for training agents. While there have been efforts to build representations for words, paragraphs and full texts, the generation of representations tailored for the specifics of conversations has not been explored. Such conversation representations could be used to find similar conversations or similar subsets of conversations, and in the context of conversations classifiers. The aim of this thesis is to propose and evaluate the benefit of conversation representations that are suitable to convey topical, structural, and behavioral aspects of the underlying conversations.

Status As of July 2018, Jeremy has proposed methods for representing conversations through recurrent and convolutional neural networks trained to predict post-hoc satisfaction of the customer. He has explored various sources of information such as words, dialog acts, and various architectures such as hierarchical models, and attention-based models. He is now focusing on contributing representations that can highlight similarities between parts of conversations, and the evaluation of proposed representations in the context of application tasks such as agent training.

Co-publications

Auguste, Jeremy, Delphine Charlet, Géraldine Damnati, Benoit Favre, and Frederic Bechet. 2018. “Evaluation automatique de la satisfaction client à partir de conversations de type « chat » par réseaux de neurones récurrents avec mécanisme d’attention.” In *TALN*.

Auguste, Jeremy, Arnaud Rey, and Benoit Favre. 2017. “Evaluation of word embeddings against cognitive processes: primed reaction times in lexical decision and naming tasks”. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, 21–26.

8.7 Manon Scholivet

Language-independent natural language processing 2017+, funded by doctoral school ED184 (33% with C. Ramisch & A. Nasr)

Summary More than 6,000 spoken and/or written languages exist in the world. The current approach to natural language processing which consists in manually annotating texts with structures and training systems to predict them is not sustainable at the scale of all languages. The previous approach to tackle this problem was to revert to unsupervised learning in order to discover structures based on raw texts or speech, but those approaches were still applied language by language. However, on one hand, there exist language families, and shared properties across languages which one may leverage to model invariants that will generalize well across languages. On the other hand, Chomsky’s theory that the brain contains universals to implement any kind of language, of which a particular language is just a configuration, has not been disproved. Those observations give confidence that it should be possible to create NLP systems that are able to process a never-seen language given the amount of supervised or unsupervised knowledge they have been exposed to. The objective of this thesis is to take advantage

of the universal dependency treebank (UD), which contains annotated syntactic trees for 60 languages in a unique representation of syntactic phenomena, in order to build a language-independent syntactic parser.

Status As of July 2018, Manon has tested the effect of including language-level features (such as subject-verb-object order) from the World Atlas of Language Structures in a delexicalized syntactic parser trained on the concatenation of all UD treebanks, and shown that it can benefit from them on seen and unseen languages. In the near future, she will relax the delexicalized input hypothesis (which assumes that a part-of-speech tagger and morphological analyzer are available for all languages) by proposing universal word embedding learning strategies, which project observed tokens to a common language-independent representation space and can be learned from raw text only. She will also look for ways of modeling invariants linked to morphological analysis in order to project morphology from character sequences to language-neural representations in a way that is generalizable to never-seen languages.

Co-publications

Scholivet, Manon, Franck Dary, Alexis Nasr, Benoit Favre, and Carlos Ramisch. 2019. "Typological Features for Multilingual Delexicalised Dependency". In *NAACL*.

Zampieri, Nicolas, Manon Scholivet, Carlos Ramisch, and Benoit Favre. 2018. "Veyn at PARSEME Shared Task 2018: Recurrent neural networks for VMWE identification". In *LAW-MWE-CxG Workshop at COLING*.

8.8 Simone Fuscone

Linguistic Underpinnings of Conversational Interpersonal Dynamics 2017+, DOC2AMU (50% with L. Prévot)

Summary Co-adaptation is a well-known phenomenon in dialog and multi-party conversations. Speakers tend to adopt the same speaking style, use the same words, in order to improve the efficiency of the communication channel. Convergence and entrainment phenomena have been shown to exist in many linguistics studies through measures of the conditional variation of parameters such as speech rate, phonetic choices, backchannel use, etc., of one speaker given the speech from another speaker at the conversation level. The objective of this thesis is to study how co-adaptation evolves during a conversation, as speakers tend to know each other's position and social roles tend to emerge. The first question is whether convergence and divergence dynamics can be demonstrated within a conversation, and how social roles and their evolution affect such phenomenon. Another interesting question is whether, as an economy, conversation participants tend to select a subset of channels for expressing co-adaptation, and whether intensity of the co-adaptation can be devised across multiple parameters. Finally, the thesis will explore co-adaptation prediction models in order to perceive and generate co-adaptation in the framework of automated agents.

Status As of July 2018, Simone has replicated a landmark work on convergence on speech rate based on data from the Switchboard corpus. He extended the protocol in order to be able to demonstrate the existence of convergence patterns within a conversation at a crude granularity. He proposed a novel methodological tool for showing co-adaptation by contrasting classifiers predicting a co-adaptation target with different set of features that include, or not the interlocutor of a conversation. He will also explore finer grained models for addressing the other questions in his thesis.

Co-publications

Fuscone, Simone, Benoit Favre, and Laurent Prévot. 2018. "Replicating Speech Rate Convergence Experiments on the Switchboard Corpus". In *4REAL workshop at LREC*.

Chapter 9

List of personal publications

This chapter contains a list of my personal publications. Self citations in the rest of the document are excluded from the general bibliography (page 105).

9.1 Book chapters

Hakkani-Tür, Dilek, Gokhan Tur, Benoit Favre, and Elizabeth Shriberg. 2011. "Finding the Structure of Documents". In *Multilingual natural language processing applications*, ed. by I. Zitouni and D.M. Bickel, 21–48. IBM Press.

Favre, Benoit. 2011. "Navigation dans les documents audio par le résumé automatique". In *Vers une recherche d'information contextuelle, assistée et personnalisée*, ed. by Patrice Bellot. Hermes.

Ji, Heng, Benoit Favre, Wen-Pin Lin, Dan Gillick, Dilek Hakkani-Tur, and Ralph Grishman. 2011. "Open-domain Multi-Document Summarization via Information Extraction: Challenges and Prospects". In *Multi-source Multilingual Information Extraction and Summarization. Lecture Notes in Computer Science*, ed. by H. Saggion, T. Poibeau, and R. Yangarber. Springer.

Ostendorf, Mari, Benoit Favre, Ralph Grishman, Dilek Hakkani-Tur, Mary Harper, Dustin Hillard, Julia B. Hirschberg, Heng Ji, Jeremy G. Kahn, Yang Liu, Evgeny Matusov, Hermann Ney, Elizabeth Shriberg, Wen Wang, and Chuck Wooters. 2008a. "Speech segmentation and spoken document processing". In *Handbook of Natural Language Processing and Machine Translation*, ed. by John McCary Joseph Olive Catlin Chrisianson. Springer.

9.2 International peer-reviewed journals

Riccardi, G., F. Bechet, M. Danieli, B. Favre, R. Gaizauskas, U. Kruschwitz, and M. Poesio. 2016. "The SENSEI Project: Making Sense of Human Conversations". Ed. by J.F. Quesada et al. *Lecture Notes on Artificial Intelligence* LNAI 9577 (): 10–33.

Liu, Shuhua, and Benoit Favre. 2013. "Understand the Global Economic Crisis: A Text Summarization Approach", 20.2:89–110.

Ji, H., J. Liu, B. Favre, D. Gillick, and D. Hakkani-Tur. 2010. "Re-ranking Summaries Based on Cross-Document Information Extraction". *Information Retrieval Technology*: 432–442.

Riedhammer, Korbinian, Benoit Favre, and Dilek Hakkani-Tur. 2010. "Long story short-Global unsupervised models for keyphrase based meeting summarization". *Speech Communication* 52 (10): 801–815.

Tur, Gokhan, Andreas Stolcke, Lynn Voss, John Dowding, Benoit Favre, Raquel Fernandez, Matthew Frampton, Michael Frandsen, Clint Frederickson, Martin Graciarena, Dilek Hakkani-Tür, Donald Kintzing, Kyle Leveque, Shane Mason, John Niekrasz, Stanley Peters, Matthew Purver, Korbinian Riedhammer, Elizabeth Shriberg, Jing Tien, Dimitra Vergyri, and Fan Yang. 2010. "The CALO Meeting Assistant System". *IEEE Transactions on Audio, Speech and Language Processing*.

Shriberg, Elizabeth, Benoit Favre, James Fung, Dilek Hakkani-Tür, and Sébastien Cuendet. 2009. "Prosodic Similarities of Dialog Act Boundaries Across Speaking Styles". Ed. by Shu-Chuan Tseng. *Language and Lingusitics Monograph Series: Linguistic Patterns in Spontaneous Speech*, no. A25: 213–239.

Guz, Umit, Benoit Favre, Dilek Hakkani-Tür, and Gokhan Tur. 2009. "Generative and Discriminative Methods using Morphological Information for Sentence Segmentation of Turkish". *IEEE Transactions on Audio, Speech and Language Processing, Special Section on Processing Morphologically Rich Languages* 17, no. 5 (): 895–903.

Ostendorf, Mari, Benoit Favre, Ralph Grishman, Dilek Hakkani-Tur, Mary Harper, Dustin Hillard, Julia B. Hirschberg, Heng Ji, Jeremy G. Kahn, Yang Liu, Evgeny Matusov, Hermann Ney, Elizabeth Shriberg, Wen Wang, and Chuck Wooters. 2008b. "Speech segmentation and spoken document processing". *Signal Processing Magazine, IEEE* 25 (3): 59–69.

Cuendet, S., D. Hakkani-Tur, E. Shriberg, J. Fung, and B. Favre. 2007a. "Cross-Genre Feature Comparisons for Spoken Sentence Segmentation". *International Journal on Semantic Computing* 1:3:335–346.

9.3 International peer-reviewed conferences

Rouvier, Mickael, and Benoit Favre. 2016a. "SENSEI-LIF at SemEval-2016 Task 4: Polarity embedding fusion for robust sentiment analysis." In *SemEval@ NAACL-HLT*, 202–208.

Favre, Benoit, Bernd Bohnet, and Dilek Hakkani-Tür. 2010a. "Evaluation of Semantic Role Labeling and Dependency Parsing of Automatic Speech Recognition Output". In *IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP), Dallas (USA)*.

Bendris, Meriem, Delphine Charlet, Benoit Favre, Geraldine Damnati, and Rémi Auguste. 2014. "Multiple-View Constrained Clustering For Unsupervised Face Identification, In TV-Broadcast". In *ICASSP2014 - Image, Video, and Multidimensional Signal Processing, (ICASSP2014 - IVMSP)*. Florence, Italy.

Rouvier, Mickael, Sebastien Delecraz, Benoit Favre, Meriem Bendris, and Frederic Bechet. 2015a. "Multimodal Embedding Fusion for Robust Speaker Role Recognition in Video Broadcast". In *IEEE ASRU*.

Rouvier, Mickael, Pierre-Michel Bousquet, and Benoit Favre. 2015b. "Speaker diarization through speaker embeddings". In *EUSIPCO*.

Delecraz, Sebastien, Alexis Nasr, Frederic Bechet, and Benoit Favre. 2017. "Correcting prepositional phrase attachments using multimodal corpora". In *Proceedings of the 15th International Conference on Parsing Technologies*, 72–77.

Gillick, Daniel, and Benoit Favre. 2009a. "A Scalable Global Model for Summarization". In *NAACL/HLT 2009 Workshop on Integer Linear Programming for Natural Language Processing*.

Favre, Benoit, Dilek Hakkani-Tür, Slav Petrov, and Dan Klein. 2008a. "Efficient Sentence Segmentation Using Syntactic Features". In *Spoken Languge Technologies (SLT), Goa (India)*.

Nasr, A., F. Béchet, J.F. Rey, B. Favre, and J. Le Roux. 2011. "MACAON: an NLP tool suite for processing word lattices". In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, 86–91. Association for Computational Linguistics.

Tafforeau, Jeremie, Frederic Bechet, Thierry Artiere, and Benoit Favre. 2016. "Joint syntactic and semantic analysis with a multitask Deep Learning Framework for Spoken Language Understanding". In *Interspeech, San Francisco (USA)*.

Bechet, Frederic, and Benoit Favre. 2013. "Asr Error Segment Localization for Spoken Recovery Strategy". In *IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP), Vancouver (Canada)*.

Xie, Shasha, Dilek Hakkani-Tür, Benoit Favre, and Yang Liu. 2009a. "Integrating Prosodic Features in Extractive Meeting Summarization". In *ASRU, Merano (Italy)*.

Trione, Jérémy, Benoit Favre, and Frédéric Béchet. 2016a. "Beyond utterance extraction: summary recombination for speech summarization". In *Interspeech, San Francisco (USA)*.

Tafforeau, Jeremie, Frederic Bechet, Benoit Favre, and Thierry Artieres. 2015a. "Lexical embedding adaptation for open-domain spoken language understanding". In *NIPS Workshop on Spoken Language Understanding (SLUNIPS)*.

Rouvier, Mickael, and Benoit Favre. 2014a. "Speaker adaptation of DNN-based ASR with i-vectors: Does it actually adapt models to speakers?" In *Interspeech, Singapore*.

Scholivet, Manon, Franck Dary, Alexis Nasr, Benoit Favre, and Carlos Ramisch. 2019. "Typological Features for Multilingual Delexicalised Dependency". In *NAACL*.

Favre, Benoit, Frédéric Béchet, and Pascal Nocéra. 2005a. "Robust Named Entity Extraction from Spoken Archives". In *Proceedings of HLT-EMNLP'05, Vancouver (Canada)*.

— . 2005b. "Mining Broadcast News data: Robust Information Extraction from Word Lattices". In *Proceeding of Eurospeech'05, Lisboa (Portugal)*.

Bechet, Frederic, Meriem Bendris, Delphine Charlet, Geraldine Damnati, Benoit Favre, Mickael Rouvier, Remi Auguste, Benjamin Bigot, Richard Dufour, Corinne Fredouille, Georges Linares, Grégory Senay, Pierre Tirilly, and Jean Martinet. 2014a. "Multimodal understanding for person recognition in video broadcasts". In *Interspeech, Singapore*.

Favre, Benoit, Géraldine Damnati, Frederic Bechet, Meriem Bendris, Delphine Charlet, Rémi Auguste, Stéphane Ayache, Benjamin Bigot, Alexandre Delteil, Richard Dufour, Corinne Fredouille, Georges Linarès, Jean Martinet, Gregory Senay, and Pierre Tirilly. 2013a. "PERCOLI: a person identification system for the 2013 REPERE challenge". In *First Workshop on Speech, Language and Audio in Multimedia (SLAM), Marseille, France*, 55–60.

Bendris, Meriem, Delphine Charlet, Gregory Senay, MinYoung Kim, Benoit Favre, Mickael Rouvier, Frederic Bechet, and Géraldine Damnati. 2015. "PERCOLATTE: A Multimodal Person Discovery System in TV Broadcast for the Medieval 2015 Evaluation Campaign". In *Mediaeval 2015 Multimedia Benchmark Workshop, Wurzen, Germany*.

Favre, Benoit, Evgeny Stepanov, Jérémy Trione, Frédéric Béchet, and Giuseppe Riccardi. 2015. "Call Centre Conversation Summarization: A Pilot Task at Multiling 2015". In *Sigdial*.

Favre, Benoit, Kyla Cheung, Siavash Kazemian, Adam Lee, Yang Liu, Cosmin Munteanu, Ani Nenkova, Dennis Ochei, Gerald Penn, Stephen Tratz, Clare Voss, and Frauke Zeller. 2013b. "Automatic Human Utility Evaluation of ASR Systems: Does WER Really Predict Performance?" In *Interspeech, Lyon (France)*.

Fuscone, Simone, Benoit Favre, and Laurent Prévot. 2018. "Replicating Speech Rate Convergence Experiments on the Switchboard Corpus". In *4REAL workshop at LREC*.

Delecraz, Sebastien, Alexis Nasr, Frederic Bechet, and Benoit Favre. 2018a. "Adding Syntactic Annotations to Flickr30k Entities Corpus for Multimodal Ambiguous Prepositional-Phrase Attachment Resolution". In *LREC*.

Delecraz, Sebastien, Leonor Becerra-Bonache, Alexis Nasr, Frederic Bechet, and Benoit Favre. 2019. "Visual Disambiguation of Prepositional Phrase Attachments: Multimodal Machine Learning for Syntactic Analysis Correction". In *International Work-Conference on Artificial Neural Networks*, 632–643. Springer.

Marzinotto, Gabriel, Geraldine Damnati, Frederic Bechet, and Benoit Favre. 2019. "Robust Semantic Parsing with Adversarial Learning for Domain Generalization". In *NAACL*.

Zampieri, Nicolas, Manon Scholivet, Carlos Ramisch, and Benoit Favre. 2018. "Veyn at PARSEME Shared Task 2018: Recurrent neural networks for VMWE identification". In *LAW-MWE-CxG Workshop at COLING*.

Auguste, Jeremy, Arnaud Rey, and Benoit Favre. 2017. "Evaluation of word embeddings against cognitive processes: primed reaction times in lexical decision and naming tasks". In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, 21–26.

Funk, Adam, Robert Gaizauskas, and Benoit Favre. 2016. "A Document Repository for Social Media and Speech Conversations". In *Language Resources and Evaluation Conference (LREC)*.

Ghannay, Sahar, Benoit Favre, Yannick Estève, and Nathalie Camelin. 2016. "Word embedding evaluation and combination". In *Language Resources and Evaluation Conference (LREC)*.

Danieli, Morena, Balamurali A R, Evgeny Stepanov, Benoit Favre, Frederic Bechet, and Giuseppe Riccardi. 2016. "Summarizing Behaviours: An Experiment on the Annotation of Call-Centre Conversations". In *Language Resources and Evaluation Conference (LREC)*.

Rouvier, Mickael, and Benoit Favre. 2016b. "Investigation of Speaker Embeddings for Cross-show Speaker Diarization". In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

R, Balamurali A, Frédéric Béchet, and Benoit Favre. 2016. "CallAn: A Tool to Analyze Call Center Conversations". In *International Workshop on Spoken Dialogue Systems (IWSDS)*.

Favre, Benoit, Mickael Rouvier, Frédéric Béchet, and Rocio Berenguer. 2016. "Speech Input for Live Performance: An Impromptu Dialogue Between the Computer and the Artist". In *International Workshop on Spoken Dialogue Systems (IWSDS)*.

Boudin, Florian, Hugo Mougard, and Benoit Favre. 2015. "Concept-based Summarization using Integer Linear Programming: From Concept Pruning to Multiple Optimal Solutions". In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2015*.

Bechet, Frederic, Benoit Favre, and Mickael Rouvier. 2015a. "Speech is silver, but silence is golden: improving speech-to-speech translation performance by slashing users input". In *Interspeech*.

Tafforeau, Jeremie, Thierry Artieres, Benoit Favre, and Frederic Bechet. 2015b. "Adapting lexical representation and OOV handling from written to spoken language with word embedding". In *Interspeech*.

Giannakopoulos, George, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. 2015. "MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations". In *Sigdial*.

Trione, Jeremy, Frederic Bechet, Benoit Favre, and Alexis Nasr. 2015. "Rapid FrameNet annotation of spoken conversation transcripts". In *Joint ACL-ISO Workshop on Interoperable Semantic Annotation*.

Rouvier, Michael, Benoit Favre, and Frederic Bechet. 2014b. "Joint Decoding of Complementary Utterances". In *Spoken Languge Technologies (SLT), Lake Tahoe (USA)*.

Bechet, Frederic, Alexis Nasr, and Benoit Favre. 2014b. "Adapting dependency parsing to spontaneous speech for open domain spoken language understanding". In *Interspeech, Singapore*.

Meriem, Bendris, Delphine Charlet, Damnati Geraldine, Benoit Favre, and Mickael Rouvier. 2014. "Scene understanding for identifying persons in TV shows: beyond face authentication". In *12th International Workshop on Content-Based Multimedia Indexing (CBMI), Klagenfurt (Austria)*.

Hong, Kai, John M. Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. "A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization". In *LREC*.

Nasr, Alexis, Frederic Bechet, Benoit Favre, Thierry Bazillon, Jose Deulofeu, and Andre Valli. 2014. "Automatically enriching spoken corpora with syntactic information for linguistic studies". In *LREC*.

Bechet, Frederic, Benoit Favre, Alexis Nasr, and Mathieu Morey. 2014c. "Retrieving the syntactic structure of erroneous ASR transcriptions for open-domain Spoken Language Understanding". In *ICASSP2014 - Speech and Language Processing (ICASSP2014 - SLTC)*. Florence, Italy.

Favre, Benoit, Mickael Rouvier, and Frederic Bechet. 2014. "Reranked aligners for interactive transcript correction". In *ICASSP2014 - Speech and Language Processing (ICASSP2014 - SLTC)*. Florence, Italy.

Bendris, Meriem, Benoit Favre, Delphine Charlet, and Geraldine Damnati. 2013. "Unsupervised Face Identification in TV Content using Audio-Visual Sources". In *11th International Workshop on Content-Based Multimedia Indexing (CBMI), Veszprém (Hungary)*.

Rouvier, Mickael, Georges Linares, Bernard Merialdo, and Benoit Favre. 2013. "Searching Segments of Interest in Single Story Web-Videos". In *Workshop on Image and Audio Analysis for Multimedia Interactive Services (WIAMIS)*.

Ayan, Necip Fazil, Arindam Mandal, Michael Frandsen, Jing Zheng, Peter Blasco, Andreas Kathol, Frederic Bechet, Benoit Favre, Alex Marin, Tom Kwiatkowski, Mari Ostendorf, Luke Zettlemoyer, Philipp Salletmayr, Julia Hirschberg, and Svetlana Stoyanchev. 2013. "Can You Give Me Another Word for Hyperbaric?: Improving Speech Translation using Targeted Clarification Questions". In *IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP), Vancouver (Canada)*.

Roux, Joseph Le, Benoit Favre, Alexis Nasr, and Seyed Abolghasem Mirroshandel. 2012. "Generative Constituent Parsing and Discriminative Dependency Reranking: Experiments on English and French". In *SP-SEM-MRL 2012*.

Lefèvre, Fabrice, Djamel Mostefa, Laurent Besacier, Yannick Estève, Matthieu Quignard, Nathalie Camelin, Benoit Favre, Bassam Jabaian, and Lina Rojas-Barahona. 2012a. "Leveraging study of robustness and portability of spoken language understanding systems across languages and domains: the PortMedia corpora". In *LREC'12, Istambul, Turkey*.

Bechet, Frederic, Benoit Favre, and Geraldine Damnati. 2012. "Detecting Person Presence in TV Shows with Linguistic and Structural Features". In *IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP), Kyoto (Japan)*.

Ralaivola, Liva, Benoit Favre, Pierre Gotab, Frédéric Béchet, and Géraldine Damnati. 2011. "Applying Multiclass Bandit algorithms to call-type classification". In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'11)*.

Dufour, Richard, and Benoit Favre. 2010. "Semi-supervised Part-of-speech Tagging in Speech Applications". In *Interspeech, Tokyo (Japan)*.

Boakye, Kofi, Benoit Favre, and Dilek Hakkani-Tür. 2009. "Any Questions? Automatic Question Detection in Meetings". In *ASRU, Merano (Italy)*.

Xie, Shasha, Benoit Favre, Dilek Hakkani-Tür, and Yang Liu. 2009b. "Leveraging Sentence Weights in Concept-based Optimization Framework for Extractive Meeting Summarization". In *Interspeech, Brighton (UK)*.

Garg, Nikhil, Benoit Favre, Korbinian Reidhammer, and Dilek Hakkani-Tür. 2009. "ClusterRank: A Graph Based Method for Meeting Summarization". In *Interspeech, Brighton (UK)*.

Favre, Benoit, and Dilek Hakkani-Tür. 2009b. "Phrase-level and Word-level Strategies for Detecting Appositions in Speech". In *Interspeech, Brighton (UK)*.

Lecouteux, Benjamin, Georges Linarès, and Benoit Favre. 2009. "Combined low level and high level features for Out-Of-Vocabulary Word detection". In *Interspeech, Brighton (UK)*.

Favre, B., D. Hakkani-Tur, and E. Shriberg. 2009c. "Syntactically-informed Models for Comma Prediction". In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan*.

Gillick, D., K. Riedhammer, B. Favre, and D. Hakkani-Tür. 2009c. "A Global Optimization Framework for Meeting Summarization". In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan*.

Tur, Gokhan, Andreas Stolcke, Lynn Voss, John Dowding, Benoit Favre, Raquel Fernandez, Matthew Frampton, Michael Frandsen, Clint Frederickson, Martin Graciarena, Dilek Hakkani-Tür, Donald Kintzing, Kyle Leveque, Shane Mason, John Niekrasz, Stanley Peters, Matthew Purver, Korbinian Riedhammer, Elizabeth Shriberg, Jing Tien, Dimitra Vergyri, and Fan Yang. 2008. "The CALO Meeting Speech Recognition and Understanding System". In *Spoken Languge Technologies (SLT), Goa (India)*.

Riedhammer, Korbinian, Benoit Favre, and Dilek Hakkani-Tür. 2008a. "A Keyphrase Based Approach to Interactive Meeting Summarization". In *Spoken Languge Technologies (SLT), Goa (India)*.

— . 2008b. "Packing the Meeting Summarization Knapsack". In *Interspeech, Brisbane (Australia)*.

Favre, B., R. Grishman, D. Hillard, H. Ji, D. Hakkani-Tur, and M. Ostendorf. 2008b. "Punctuating Speech for Information Extraction". In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Las Vegas, Nevada*.

Favre, Benoit, Jean-François Bonastre, and Patrice Bellot. 2007c. "An Interactive Timeline for Speech Database Browsing". In *Interspeech 2007, Antwerp (Belgium)*.

Cuendet, Sebastien, Elizabeth Shriberg, Benoit Favre, James Fung, and Dilek Hakkani-Tur. 2007b. "An Analysis of Sentence Segmentation Features for Broadcast News, Broadcast Conversations, and Meetings". In *Proceedings of SIGIR 2007, Searching Spontaneous Conversational Speech (SSCS) workshop*, 37–43.

Favre, Benoit, Patrice Bellot, and Jean-François Bonastre. 2004a. "Information retrieval on mixed written and spoken documents". In *RIAO, Avignon (France)*, 826–835.

9.4 National peer-reviewed conferences

Auguste, Jeremy, Delphine Charlet, Géraldine Damnati, Benoit Favre, and Frederic Bechet. 2018. "Evaluation automatique de la satisfaction client à partir de conversations de type « chat » par réseaux de neurones récurrents avec mécanisme d'attention." In *TALN*.

Magallon, Thibault, Frederic Bechet, and Benoit Favre. 2018a. "Détection d'erreurs dans des transcriptions OCR de documents historiques par réseaux de neurones récurrents multi-niveau." In *TALN*.

Roux, Joseph Le, Benoit Favre, Seyed Abolghasem Mirroshandel, and Alexis Nasr. 2011. "Modèles génératif et discriminant en analyse syntaxique : expériences sur le corpus arboré de Paris 7". In *TALN'11, Montpellier (France)*.

Delecraz, Sebastien, Leonor Becerra-Bonache, Benoit Favre, Alexis Nasr, and Frederic Bechet. 2018b. "Correction automatique d'attachements prépositionnels par utilisation de traits visuels." In *TALN*.

Magallon, Thibault, Frederic Bechet, and Benoit Favre. 2018b. "Fusion multimodale image/texte par réseaux de neurones profonds pour la classification de documents imprimés." In *CORIA*.

Favre, Benoit, Frédéric Béchet, Géraldine Damnati, and Délphine Charlet. 2017. "Apprentissage d'agents conversationnels pour la gestion de relations clients". In *24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*.

Godbert, Elisabeth, and Benoit Favre. 2017. "Détection de coréférences de bout en bout en français". In *24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, 52.

Delecraz, Sebastien, Frederic Bechet, Benoit Favre, and Mickael Rouvier. 2016. "Fusion d'espaces de représentations multimodaux pour la reconnaissance du rôle du locuteur dans des documents télévisuels". In *Actes de la conférence JEP 2016*. Paris, France: AFCP.

Trione, Jérémy, Benoit Favre, and Frederic Bechet. 2016b. "Détection de concepts pertinents pour le résumé automatique de conversations par recombinaison de patrons". In *Actes de la conférence TALN 2016*. Paris, France: ATALA.

Bechet, Frederic, Meriem Bendris, Delphine Charlet, Geraldine Damnati, Benoit Favre, Mickael Rouvier, Remi Auguste, Benjamin Bigot, Richard Dufour, Corinne Fredouille, Georges Linares, Jean Martinet, Gregory Senay, and Pierre Trilly. 2015b. "Identification de personnes dans des flux multimédia". In *CORIA*.

Béchet, Frédéric, and Benoit Favre. 2014. "Détection et caractérisation d'erreurs dans des transcriptions automatiques pour des systèmes de traduction parole-parole". In *Actes de la conférence JEP 2014*. Le Mans, France: AFCP.

Rouvier, Mickael, Benoît Favre, and Frédéric Béchet. 2014c. "Correction interactive de transcriptions de parole par fusion de phrases". In *Actes de la conférence JEP 2014*. Le Mans, France: AFCP.

Béchet, Frédéric, Rémi Auguste, Stéphane Ayache, Delphine Charlet, Géraldine Damnati, Benoît Favre, Corinne Fredouille, Christophe Lévy, Georges Linarès, and Jean Martinet. 2012. "Percol0 - un système multimodal de détection de personnes dans des documents vidéo (Percol0 - A multimodal person detection system in video documents) [in French]". In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 1: JEP*, 553–560. Grenoble, France: ATALA/AFCP.

Lefèvre, Fabrice, Djamel Mostefa, Laurent Besacier, Yannick Estève, Matthieu Quignard, Nathalie Camelin, Benoit Favre, Bassam Jabaian, and Lina Rojas-Barahona. 2012b. "Robustesse et portabilités multilingue et multi-domaines des systèmes de compréhension de la parole : les corpus du projet PortMedia (Robustness and portability of spoken language understanding systems among languages and domains : the PORTMEDIA project) [in French]". In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 1: JEP*, 779–786. Grenoble, France: ATALA/AFCP.

Favre, Benoit, Jean-François Bonastre, Patrice Bellot, and François Capman. 2006b. "Accès aux connaissances orales par le résumé automatique". In *EGC'06, Lille (France)*.

Favre, Benoit, Jean-François Bonastre, and Patrice Bellot. 2004b. "Recherche d'information dans un mélange de documents écrits et parlés". In *Journées d'Etude de la Parole (JEP), Fèz (Morocco)*.

9.5 Other publications

Gillick, Daniel, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009b. "The ICSI/UTD Summarization System at TAC 2009". In *Proc. of the Text Analysis Conference workshop, Gaithersburg, MD (USA)*.

Hmida, Firas, and Benoit Favre. 2011. "LIF at TAC Multiling: Towards a Truly Language Independent Summarizer". In *Proc. of the Text Analysis Conference workshop, Gaithersburg, MD (USA)*.

Gillick, Daniel, Benoit Favre, and Dilek Hakkani-Tur. 2008. "The ICSI Summarization System at TAC 2008". In *Proc. of the Text Analysis Conference workshop, Gaithersburg, MD (USA)*.

Favre, B., L. Gillard, and J.M. Torres-Moreno. 2007b. "The LIA summarization system at DUC-2007". In *Proceedings of the Document Understanding Workshop (DUC)*.

Favre, Benoit, Frédéric Béchet, Patrice Bellot, Florian Boudin, Marc El-Bèze, Laurent Gillard, Guy Lapalme, and Juan-Manuel Torres-Moreno. 2006a. "The LIA-Thales summarization system at DUC-2006". In *Document Understanding Conference Workshop, HLT-NAACL'06, New York (USA)*.

Favre, Benoit, and Bernd Bohnet. 2010b. "The UMUS system for named entity generation at GREC 2010". In *International Natural Language Generation Conference (INLG)*.

Favre, Benoit, and Bernd Bonhet. 2009a. "ICSI-CRF: The Generation of References to the Main Subject and Named Entities Using Conditional Random Fields". In *ACL-IJCNLP*.

Rouvier, Mickael, Benoit Favre, and Balamurali Andiyakkal Rajendran. 2015c. "TALEP @ DEFT'15 : Le plus cooool des systèmes d'analyse de sentiment". In *Actes de la 11e Défi Fouille de Texte*, 97–103. Caen, France: Association pour le Traitement Automatique des Langues.

Giannakopoulos, George, Mahmoud El-Haj, Benoit Favre, Marianna Litvak, Josef Steinberger, and Vasudeva Varma. 2011. "TAC2011 MultiLing Pilot Overview". In *Proc. of the Text Analysis Conference workshop, Gaithersburg, MD (USA)*.

Bogaerts, Louisa, Ana Franco, Benoit Favre, and Arnaud Rey. 2016. "Speech onset latencies as an online measure of regularity extraction". In *Poster at Implicit Learning Seminar, Lancaster (UK)*.

Stepanov, E., B. Favre, F. Alam, S. Chowdhury, K. Singla, J. Trione, F. Bechet, and G. Riccardi. 2015. "Automatic Summarization of Call-Centre Conversations". In *IEEE ASRU Demo*.

Bohnet, Bernd, Simon Mille, Benoît Favre, and Leo Wanner. 2011. "StuMaBa: from deep representation to surface". In *Proceedings of the 13th European workshop on natural language generation*, 232–235. Association for Computational Linguistics.

Favre, Benoit. 2007. "Résumé automatique de parole pour un accès efficace aux bases de données audio". PhD thesis, University of Avignon (France).

— . 2003. "Indexation multimédia : caractérisation du déséquilibre entre les modalités texte et parole". PhD thesis, University of Avignon (France).

Bibliography

This bibliography excludes self citations which are listed in Chapter 9.

Agirre, Eneko, Timothy Baldwin, and David Martinez. 2008. "Improving parsing and PP attachment performance with sense information". *proceedings of ACL-08: HLT*: 317–325.

Allauzen, Cyril, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. "OpenFst: A general and efficient weighted finite-state transducer library". In *International Conference on Implementation and Application of Automata*, 11–23. Springer.

Ammar, Waleed, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016a. "Many Languages, One Parser". *Transactions of the Association of Computational Linguistics* 4 (1): 431–444.

Ammar, Waleed, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016b. "Massively multilingual word embeddings". *arXiv preprint arXiv:1602.01925*.

Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. "Deep speech 2: End-to-end speech recognition in english and mandarin". In *International Conference on Machine Learning*, 173–182.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre. 2017. "Learning bilingual word embeddings with (almost) no bilingual data". In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:451–462.

Artieres, Thierry, et al. 2010. "Neural conditional random fields". In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 177–184.

Awasthi, Pranjal, Delip Rao, and Balaraman Ravindran. 2006. "Part of speech tagging and chunking with hmm and crf". *Proceedings of NLP Association of India (NLPAI) Machine Learning Contest 2006*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. "Neural machine translation by jointly learning to align and translate". *arXiv preprint arXiv:1409.0473*.

Bahdanau, Dzmitry, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. "End-to-end attention-based large vocabulary speech recognition". In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 4945–4949. IEEE.

Banerjee, Satanjeev, and Alon Lavie. 2005. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72.

Barrett, David, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. 2018. "Measuring abstract reasoning in neural networks". In *International Conference on Machine Learning*, 511–520.

Béchet, Frédéric, and Christian Raymond. 2018. "Is ATIS too shallow to go deeper for benchmarking Spoken Language Understanding models?" In *InterSpeech*. Hyderabad, India.

Bell, Robert M, and Yehuda Koren. 2007. "Lessons from the Netflix prize challenge." *SiGKDD Explorations* 9 (2): 75–79.

Bellegarda, Jerome R. 2004. "Statistical language model adaptation: review and perspectives". *Speech communication* 42 (1): 93–108.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. "A neural probabilistic language model". *Journal of machine learning research* 3 (Feb): 1137–1155.

Bennett, James, Stan Lanning, et al. 2007. "The netflix prize". In *Proceedings of KDD cup and workshop*, 2007:35. New York, NY, USA.

Benoist, Thierry, Bertrand Estellon, Frédéric Gardi, Romain Megel, and Karim Nouioua. 2011. "Localsolver 1. x: a black-box local-search solver for 0-1 programming". *4OR* 9 (3): 299.

Benson, Buster. 2017. *Cognitive bias codex*. https://www.visualcapitalist.com/wp-content/uploads/2017/09/cognitive-bias-infographic.html.

Berger, Adam L, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. "A maximum entropy approach to natural language processing". *Computational linguistics* 22 (1): 39–71.

Berghen, Frank Vanden, and Hugues Bersini. 2005. "CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm". *Journal of computational and applied mathematics* 181 (1): 157–175.

Blei, David M, Andrew Y Ng, and Michael I Jordan. 2003. "Latent dirichlet allocation". *Journal of machine Learning research* 3 (Jan): 993–1022.

Bod, Rens. 2007. "Is the end of supervised parsing in sight?" In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 400–407.

Bohnet, Bernd. 2010. "Very high accuracy and fast dependency parsing is not a contradiction". In *Proceedings of the 23rd international conference on computational linguistics*, 89–97. Association for Computational Linguistics.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. "Enriching word vectors with subword information". *arXiv preprint arXiv:1607.04606*.

Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. "Freebase: a collaboratively created graph database for structuring human knowledge". In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. AcM.

Bradbury, James, Stephen Merity, Caiming Xiong, and Richard Socher. 2016. "Quasi-Recurrent Neural Networks". *CoRR* abs/1611.01576. arXiv: 1611.01576.

Bradley, James V. 1958. "Complete counterbalancing of immediate sequential effects in a Latin square design". *Journal of the American Statistical Association* 53 (282): 525–528.

Braud, Chloé, and Anders Søgaard. 2017. "Is writing style predictive of scientific fraud?" In *Proceedings of the Workshop on Stylistic Variation*, 37–42.

Brill, Eric. 1992. "A simple rule-based part of speech tagger". In *Proceedings of the third conference on Applied natural language processing*, 152–155. Association for Computational Linguistics.

Cai, Deng, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. "Fast and Accurate Neural Word Segmentation for Chinese". In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2:608–615.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, and Omar F Zaidan. 2011. "Findings of the 2011 workshop on statistical machine translation". In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 22–64. Association for Computational Linguistics.

Carbonell, Jaime, and Jade Goldstein. 1998. "The use of MMR, diversity-based reranking for reordering documents and producing summaries". In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 335–336. ACM.

Chen, Danqi, and Christopher Manning. 2014. "A fast and accurate dependency parser using neural networks". In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 740–750.

Chen, Xinchi, Xipeng Qiu, Jingxiang Jiang, and Xuanjing Huang. 2015. "Gaussian Mixture Embeddings for Multiple Word Prototypes". *arXiv preprint arXiv:1511.06246*.

Chomsky, Noam. 1995. *The Minimalist Program*. Current studies in linguistics series. Cambridge, MA, USA: MIT Press.

Chung, Joon Son, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2016. "Lip reading sentences in the wild". *arXiv preprint arXiv:1611.05358* 2.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. "Empirical evaluation of gated recurrent neural networks on sequence modeling". *arXiv preprint arXiv:1412.3555*.

Collins, Michael. 2002. "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms". In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 1–8. Association for Computational Linguistics.

Collins, Mickael. 2005. *Log-Linear Models*.

Collobert, Ronan, Christian Puhrsch, and Gabriel Synnaeve. 2016. "Wav2letter: an end-to-end convnet-based speech recognition system". *arXiv preprint arXiv:1609.03193*.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. "Natural language processing (almost) from scratch". *Journal of Machine Learning Research* 12 (Aug): 2493–2537.

Comrie, Bernard, and David Gil. 2005. *The world atlas of language structures*. Vol. 1. Oxford University Press.

Conneau, Alexis, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. "Word translation without parallel data". *arXiv preprint arXiv:1710.04087*.

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-vector networks". *Machine learning* 20 (3): 273–297.

Cowles, Michael, and Caroline Davis. 1982. "On the origins of the. 05 level of statistical significance." *American Psychologist* 37 (5): 553.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. "Online passive-aggressive algorithms". *Journal of Machine Learning Research* 7 (Mar): 551–585.

Crystal, David, and Robert Henry Robins. 2018. "Language".

Dang, Hoa Trang, and Karolina Owczarzak. 2008. "Overview of the TAC 2008 Update Summarization Task." In *TAC*.

Daumé, Hal, John Langford, and Daniel Marcu. 2009. "Search-based structured prediction". *Machine learning* 75 (3): 297–325.

Dauphin, Yann N., Angela Fan, Michael Auli, and David Grangier. 2016. "Language Modeling with Gated Convolutional Networks". *CoRR* abs/1612.08083. arXiv: 1612.08083.

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. "Indexing by latent semantic analysis". *Journal of the American society for information science* 41 (6): 391.

Dehouck, Mathieu, and Pascal Denis. 2017. "Delexicalized word embeddings for cross-lingual dependency parsing". In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1:241–250.

Deng, Li, and Yang Liu. 2018. *Deep Learning in Natural Language Processing*. Springer.

Dozat, Timothy, Peng Qi, and Christopher D Manning. 2017. "Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task". *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*: 20–30.

Dror, Rotem, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. "The hitchhiker's guide to testing statistical significance in natural language processing". In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1383–1392.

Dufour, Richard, Yannick Esteve, and Paul Deléglise. 2011. "Investigation of spontaneous speech characterization applied to speaker role recognition". In *Twelfth Annual Conference of the International Speech Communication Association*.

Duong, Long, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. "Multilingual training of crosslingual word embeddings". In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1:894–904.

Dyer, Chris, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. "Transition-Based Dependency Parsing with Stack Long Short-Term Memory". In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1:334–343.

Efron, Bradley, and Charles Stein. 1981. "The jackknife estimate of variance". *The Annals of Statistics*: 586–596.

Emerson, E Allen, and Charanjit S Jutla. 1991. "Tree automata, mu-calculus and determinacy". In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, 368–377. IEEE.

Erkan, Günes, and Dragomir R Radev. 2004. "Lexrank: Graph-based lexical centrality as salience in text summarization". *Journal of Artificial Intelligence Research* 22:457–479.

Faghri, Fartash, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. "VSE++: Improved Visual-Semantic Embeddings". *arXiv preprint arXiv:1707.05612*.

Favre, Benoit, Dilek Hakkani-Tür, and Sebastien Cuendet. 2007c. *Icsiboost*. https://github.com/benob/icsiboost.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. 2017. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". *CoRR* abs/1703.03400. arXiv: 1703.03400.

Fülöp, Zoltán, and Heiko Vogler. 2009. "Weighted tree automata and tree transducers". In *Handbook of Weighted Automata*, 313–403. Springer.

Galibert, Olivier, Sophie Rosset, Cyril Grouin, Pierre Zweigenbaum, and Ludovic Quintard. 2011. "Structured and Extended Named Entity Evaluation in Automatic Speech Transcriptions." In *IJCNLP*, 518–526.

Gallo, Giorgio, Peter L Hammer, and Bruno Simeone. 1980. "Quadratic knapsack problems". In *Combinatorial optimization*, 132–149. Springer.

Gao, Yang, Ji Lin, Fisher Yu, Sergey Levine, Trevor Darrell, et al. 2018. "Reinforcement learning from imperfect demonstrations". *arXiv preprint arXiv:1802.05313*.

Gillick, Dan, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. "Multilingual Language Processing From Bytes". In *Proceedings of NAACL-HLT*, 1296–1306.

Gimpel, Kevin, Dhruv Batra, Chris Dyer, Gregory Shakhnarovich, and Virginia Tech. 2013. "A systematic exploration of diversity in machine translation". In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, vol. 3.

Goldberg, Yoav. 2016. "A primer on neural network models for natural language processing". *Journal of Artificial Intelligence Research* 57:345–420.

Gollub, Tim, Benno Stein, Steven Burrows, and Dennis Hoppe. 2012. "TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments". In *9th International Workshop on Text-based Information Retrieval (TIR 12) at DEXA*, ed. by A Min Tjoa, Stephen Liddle, Klaus-Dieter Schewe, and Xiaofang Zhou, 151–155. Los Alamitos, California: IEEE.

Gong, Yichen, Heng Luo, and Jian Zhang. 2017. "Natural Language Inference over Interaction Space". *CoRR* abs/1709.04348. arXiv: 1709.04348.

Gong, Yihong, and Xin Liu. 2001. "Generic text summarization using relevance measure and latent semantic analysis". In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 19–25. ACM.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative adversarial nets". In *Advances in neural information processing systems*, 2672–2680.

Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. "Hybrid speech recognition with deep bidirectional LSTM". In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, 273–278. IEEE.

Greenberg, Joseph H. 1963. *Universals of Human Language*. Cambridge, MA, USA: MIT Press.

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. "Improved training of wasserstein gans". In *Advances in Neural Information Processing Systems*, 5769–5779.

Hakkani-Tür, Dilek, Frédéric Béchet, Giuseppe Riccardi, and Gokhan Tur. 2006. "Beyond ASR 1-best: Using word confusion networks in spoken language understanding". *Computer Speech & Language* 20 (4): 495–514.

Hamilton, William L, Jure Leskovec, and Dan Jurafsky. 2016. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change". In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:1489–1501.

Hannun, Awni, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. "Deep speech: Scaling up end-to-end speech recognition". *arXiv preprint arXiv:1412.5567*.

Harris, Zellig S. 1954. "Distributional structure". *Word* 10 (2-3): 146–162.

Haug, Charlotte J. 2015. "Peer-review fraud—hacking the scientific publication process". *New England Journal of Medicine* 373 (25): 2393–2395.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. "Deep Residual Learning for Image Recognition". *CoRR* abs/1512.03385. arXiv: 1512.03385.

Hermansky, Hynek, Daniel PW Ellis, and Sangita Sharma. 2000. "Tandem connectionist feature extraction for conventional HMM systems". In *icassp*, 1635–1638. IEEE.

Howard, Jeremy, and Sebastian Ruder. 2018. "Universal Language Model Fine-tuning for Text Classification". In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339.

Huang, Gao, Zhuang Liu, and Kilian Q. Weinberger. 2016. "Densely Connected Convolutional Networks". *CoRR* abs/1608.06993. arXiv: 1608.06993.

Huang, Xuedong D, Yasuo Ariki, and Mervyn A Jack. 1990. "Hidden Markov models for speech recognition".

Jaderberg, Max, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. 2016. "Reinforcement learning with unsupervised auxiliary tasks". *arXiv preprint arXiv:1611.05397*.

Jannet, Mohamed Ameur Ben, Olivier Galibert, Martine Adda-Decker, and Sophie Rosset. 2015. "How to evaluate ASR output for named entity recognition?" In *Sixteenth Annual Conference of the International Speech Communication Association*.

Jelinek, Frederick. 1997. *Statistical methods for speech recognition*. MIT press.

Jiang, Yong, Wenjuan Han, and Kewei Tu. 2016. "Unsupervised neural dependency parsing". In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 763–771.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. "Bag of tricks for efficient text classification". *arXiv preprint arXiv:1607.01759*.

Jozefowicz, Rafal, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. "Exploring the limits of language modeling". *arXiv preprint arXiv:1602.02410*.

Józefowicz, Rafal, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. "Exploring the Limits of Language Modeling". *CoRR* abs/1602.02410. arXiv: 1602.02410.

Karp, Richard M. 1972. "Reducibility among combinatorial problems". In *Complexity of computer computations*, 85–103. Springer.

Khuller, Samir, Anna Moss, and Joseph Seffi Naor. 1999. "The budgeted maximum coverage problem". *Information Processing Letters* 70 (1): 39–45.

Kim, Yoon, Carl Denton, Luong Hoang, and Alexander M Rush. 2017. "Structured attention networks". *arXiv preprint arXiv:1702.00887*.

Kirchhoff, Katrin, Yik-Cheung Tam, Colleen Richey, and Wen Wang. 2015. "Morphological Modeling for Machine Translation of English-Iraqi Arabic Spoken Dialogs". In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 995–1000.

Kiss, Tibor, and Jan Strunk. 2006. "Unsupervised multilingual sentence boundary detection". *Computational Linguistics* 32 (4): 485–525.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. "Opennmt: Open-source toolkit for neural machine translation". *arXiv preprint arXiv:1701.02810*.

Koehn, Philipp. 2009. *Statistical machine translation*. Cambridge University Press.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. "Moses: Open source toolkit for statistical machine translation". In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, 177–180. Association for Computational Linguistics.

Konkol, Michal, and Miloslav Konopík. 2015. "Segment representations in named entity recognition". In *International Conference on Text, Speech, and Dialogue*, 61–70. Springer.

Krizhevsky, Alex, I Sutskever, and G Hinton. 2014. "ImageNet Classification with Deep Convolutional Neural". In *Neural Information Processing Systems*, 1–9.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "Imagenet classification with deep convolutional neural networks". In *Advances in neural information processing systems*, 1097–1105.

Kupiec, Julian. 1992. "Robust part-of-speech tagging using a hidden Markov model". *Computer Speech & Language* 6 (3): 225–242.

Kupiec, Julian, Jan Pedersen, and Francine Chen. 1995. "A trainable document summarizer". In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 68–73. ACM.

Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. 2016. "Adversarial examples in the physical world". *arXiv preprint arXiv:1607.02533*.

Lafferty, John, Andrew McCallum, and Fernando CN Pereira. 2001. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data".

Le Roux, Joseph, Antoine Rozenknop, and Matthieu Constant. 2014. "Syntactic parsing and compound recognition via dual decomposition: application to French". In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 1875–1885.

Lecorvé, Gwénolé, Guillaume Gravier, and Pascale Sébillot. 2008. "An unsupervised web-based topic language model adaptation method". In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 5081–5084. IEEE.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86 (11): 2278–2324.

Leek, Jeff, Blakeley B McShane, Andrew Gelman, David Colquhoun, Michèle B Nuijten, and Steven N Goodman. 2017. *Five ways to fix statistics*.

Leggetter, Christopher J, and Philip C Woodland. 1995. "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models". *Computer speech & language* 9 (2): 171–185.

Levy, Omer, and Yoav Goldberg. 2014. "Dependency-based word embeddings". In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2:302–308.

Li, Jiwei, and Dan Jurafsky. 2015. "Do Multi-Sense Embeddings Improve Natural Language Understanding?" In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1722–1732.

Li, Yang, and Tao Yang. 2018. "Word Embedding for Understanding Natural Language: A Survey". In *Guide to Big Data Applications*, 83–104. Springer.

Liang, Percy, and Jacob Abernethy. 2010. *MLComp: a free website for objectively comparing machine learning programs*. http://mlcomp.org.

Lin, Chin-Yew. 2004. "Rouge: A package for automatic evaluation of summaries". *Text Summarization Branches Out*.

Lin, Hui, and Jeff Bilmes. 2011. "A class of submodular functions for document summarization". In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 510–520. Association for Computational Linguistics.

Lipton, Zachary C. 2016. "The mythos of model interpretability". *arXiv preprint arXiv:1606.03490*.

Liu, Chenxi, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. 2018. "Progressive neural architecture search". In *Proceedings of the European Conference on Computer Vision (ECCV)*, 19–34.

Liu, Dong C, and Jorge Nocedal. 1989. "On the limited memory BFGS method for large scale optimization". *Mathematical programming* 45 (1-3): 503–528.

Lopez-Paz, David, et al. 2017. "Gradient Episodic Memory for Continual Learning". In *Advances in Neural Information Processing Systems*, 6470–6479.

Luong, Minh-Thang, Hieu Pham, and Christopher D Manning. 2015. "Effective approaches to attention-based neural machine translation". *arXiv preprint arXiv:1508.04025*.

Lupyan, Gary. 2016. "The Centrality of Language in Human Cognition". *Language Learning* 66 (3): 516–553.

Lynn, Teresa, Jennifer Foster, Mark Dras, and Lamia Tounsi. 2014. "Cross-lingual transfer parsing for low-resourced languages: An Irish case study". In *Proceedings of the First Celtic Language Technology Workshop*, 41–49.

Ma, Xuezhe, and Eduard Hovy. 2016. "End-to-end sequence labeling via bi-directional lstm-cnns-crf". *arXiv preprint arXiv:1603.01354*.

Machacek, Matous, and Ondrej Bojar. 2014. "Results of the WMT14 metrics shared task". In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 293–301.

Manning, Christopher D. 2011. "Part-of-speech tagging from 97% to 100%: is it time for some linguistics?" In *International conference on intelligent text processing and computational linguistics*, 171–189. Springer.

Manning, Christopher D, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

Markowitz, David M, and Jeffrey T Hancock. 2016. "Linguistic obfuscation in fraudulent science". *Journal of Language and Social Psychology* 35 (4): 435–445.

Martin, James H, and Daniel Jurafsky. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall.

Matrouf, Driss, J-F Bonastre, and Corinne Fredouille. 2006. "Effect of speech transformation on impostor acceptance". In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 1:I–I. IEEE.

McCann, Bryan, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. "The Natural Language Decathlon: Multitask Learning as Question Answering". *arXiv preprint arXiv:1806.08730*.

McDonald, Ryan. 2007. "A study of global inference algorithms in multi-document summarization". In *European Conference on Information Retrieval*, 557–564. Springer.

McDonald, Ryan, Koby Crammer, and Fernando Pereira. 2005a. "Online large-margin training of dependency parsers". In *Proceedings of the 43rd annual meeting on association for computational linguistics*, 91–98. Association for Computational Linguistics.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005b. "Non-projective dependency parsing using spanning tree algorithms". In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 523–530. Association for Computational Linguistics.

Meguerditchian, Adrien, and Jacques Vauclair. 2008. "Vocal and gestural communication in nonhuman primates and the question of the origin of language".

Michalon, Olivier, Corentin Ribeyre, Marie Candito, and Alexis Nasr. 2016. "Deeper syntax for better semantic parsing". In *Coling 2016 - 26th International Conference on Computational Linguistics*. Osaka, Japan.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. "Distributed representations of words and phrases and their compositionality". In *Advances in neural information processing systems*, 3111–3119.

Mikolov, Tomas, and Geoffrey Zweig. 2012. "Context dependent recurrent neural network language model." *SLT* 12:234–239.

Mirroshandel, Seyed Abolghasem, Alexis Nasr, and Joseph Le Roux. 2012. "Semi-supervised dependency parsing using lexical affinities". In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 777–785. Association for Computational Linguistics.

Murray, Gabriel, Thomas Kleinbauer, Peter Poller, Tilman Becker, Steve Renals, and Jonathan Kilgour. 2009. "Extrinsic summarization evaluation: A decision audit task". *ACM Transactions on Speech and Language Processing (TSLP)* 6 (2): 2.

Nair, Vinod, and Geoffrey E Hinton. 2010. "Rectified linear units improve restricted boltzmann machines". In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Nallapati, Ramesh, Bing Xiang, and Bowen Zhou. 2016. "Sequence-to-Sequence RNNs for Text Summarization". *CoRR* abs/1602.06023. arXiv: 1602.06023.

Naseem, Tahira, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. "Using universal linguistic knowledge to guide grammar induction". In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1234–1244. Association for Computational Linguistics.

Navigli, Roberto. 2009. "Word sense disambiguation: A survey". *ACM Computing Surveys (CSUR)* 41 (2): 10.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. "Universal Dependencies v1: A Multilingual Treebank Collection." In *LREC*.

Noreen, Eric W. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Oord, Aäron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. "WaveNet: A Generative Model for Raw Audio". *CoRR* abs/1609.03499. arXiv: 1609.03499.

Oyeka, Ikewelugo Cyprian Anaene, and Godday Uwawunkonye Ebuh. 2012. "Modified Wilcoxon signed-rank test". *Open Journal of Statistics* 2 (02): 172.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. "BLEU: a method for automatic evaluation of machine translation". In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. "GloVe: Global Vectors for Word Representation". In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. "Deep Contextualized Word Representations". In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1:2227–2237.

Petridis, Stavros, Themos Stafylakis, Pingchuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. 2018. "End-to-end Audiovisual Speech Recognition". *arXiv preprint arXiv:1802.06424*.

Petrov, Slav, and Dan Klein. 2007. "Improved inference for unlexicalized parsing". In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 404–411.

Pitman, Edwin JG. 1937. "Significance tests which may be applied to samples from any populations". *Supplement to the Journal of the Royal Statistical Society* 4 (1): 119–130.

Pitz, Michael, and Hermann Ney. 2005. "Vocal tract normalization equals linear transformation in cepstral space". *IEEE Transactions on Speech and Audio Processing* 13 (5): 930–944.

Price, Patti J. 1990. "Evaluation of spoken language systems: The ATIS domain". In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Qiu, Lin, Kewei Tu, and Yong Yu. 2016. "Context-Dependent Sense Embedding". In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 183–191.

Qiu, Siyu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. "Co-learning of Word Representations and Morpheme Representations." In *COLING*, 141–150.

Quattoni, Ariadna, Sybor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. 2007. "Hidden conditional random fields". *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10: 1848–1852.

Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. "Dependency treelet translation: Syntactically informed phrasal SMT". In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 271–279. Association for Computational Linguistics.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. "Language models are unsupervised multitask learners". *OpenAI Blog* 1:8.

Recht, Benjamin, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2018. "Do CIFAR-10 Classifiers Generalize to CIFAR-10?" *arXiv preprint arXiv:1806.00451*.

Redmon, Joseph, and Ali Farhadi. 2016. "YOLO9000: better, faster, stronger". *arXiv preprint* 1612.

Riezler, Stefan, and John T Maxwell. 2005. "On some pitfalls in automatic evaluation and significance testing for MT". In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 57–64.

Rohrmeier, Martin, Willem Zuidema, Geraint A Wiggins, and Constance Scharff. 2015. "Principles of structure building in music, language and animal song". *Philosophical Transactions of the Royal Society B: Biological Sciences* 370 (1664): 20140097.

Rush, Alexander M, Sumit Chopra, and Jason Weston. 2015. "A neural attention model for abstractive sentence summarization". *arXiv preprint arXiv:1509.00685*.

Ryang, Seonggi, and Takeshi Abekawa. 2012. "Framework of automatic text summarization using reinforcement learning". In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 256–265. Association for Computational Linguistics.

Sankaran, Baskaran. 2010. "A survey of unsupervised grammar induction". *Manuscript, Simon Fraser University* 47.

Schapire, Robert E, and Yoram Singer. 2000. "BoosTexter: A boosting-based system for text categorization". *Machine learning* 39 (2-3): 135–168.

— . 1999. "Improved boosting algorithms using confidence-rated predictions". *Machine learning* 37 (3): 297–336.

Schwenk, Holger, and Jean-Luc Gauvain. 2002. "Connectionist language modeling for large vocabulary continuous speech recognition". In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, 1:I–765. IEEE.

See, Abigail, Peter J. Liu, and Christopher D. Manning. 2017. "Get To The Point: Summarization with Pointer-Generator Networks". *CoRR* abs/1704.04368. arXiv: 1704.04368.

Settles, Burr. 2012. "Active learning". *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6 (1): 1–114.

Simonsohn, Uri. 2013. "Just post it: The lesson from two cases of fabricated data detected by statistics alone". *Psychological science* 24 (10): 1875–1888.

Singer, Yoram. 2010. "Composite Objective Optimization and Learning for Massive Datasets".

SNOVER, M, B DORR, R SCHWARTZ, J MAKHOUL, L MICCIULA, and R WEISCHEDEL. 2005. *A study of translation error rate with targeted human annotation. Rapport technique LAMP-TR-126*. Tech. rep. CS-TR-4755, UMIACS-TR-2005-58, University of Maryland, College Park and BBN Technologies.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. "A study of translation edit rate with targeted human annotation". In *Proceedings of association for machine translation in the Americas*, vol. 200. 6.

Søgaard, Anders, Anders Johannsen, Barbara Plank, Dirk Hovy, and Héctor Martínez Alonso. 2014. "What's in a p-value in NLP?" In *Proceedings of the eighteenth conference on computational natural language learning*, 1–10.

Sönmez, M Kemal, Larry Heck, Mitchel Weintraub, and Elizabeth Shriberg. 1997. "A lognormal tied mixture model of pitch for prosody based speaker recognition". In *Fifth European Conference on Speech Communication and Technology*.

Specia, Lucia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. "A shared task on multimodal machine translation and crosslingual image description". In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, 2:543–553.

Srivastava, Rupesh K, Klaus Greff, and Jürgen Schmidhuber. 2015a. "Training very deep networks". In *Advances in neural information processing systems*, 2377–2385.

Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber. 2015b. "Highway Networks". *CoRR* abs/1505.00387. arXiv: 1505.00387.

Stephens, Kenneth R. 2004. "What has the Loebner contest told us about conversant systems?" *Cambridge Center for Behavioral Studies*.

Stolcke, Andreas, Yochai Konig, and Mitchel Weintraub. 1997. "Explicit word error minimization in n-best list rescoring." In *Eurospeech*, 97:163–166.

Sukhbaatar, Sainbayar, Jason Weston, Rob Fergus, et al. 2015. "End-to-end memory networks". In *Advances in neural information processing systems*, 2440–2448.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. "Sequence to sequence learning with neural networks". In *Advances in neural information processing systems*, 3104–3112.

Toutanova, Kristina, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. "Feature-rich part-of-speech tagging with a cyclic dependency network". In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 173–180. Association for Computational Linguistics.

Trentin, Edmondo, and Marco Gori. 2001. "A survey of hybrid ANN/HMM models for automatic speech recognition". *Neurocomputing* 37 (1-4): 91–126.

Tur, Gokhan, Jerry Wright, Allen Gorin, Giuseppe Riccardi, and Dilek Hakkani-Tür. 2002. "Improving spoken language understanding using word confusion networks". In *Seventh International Conference on Spoken Language Processing*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need". *CoRR* abs/1706.03762. arXiv: 1706.03762.

Vincent, Pascal, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion". *Journal of machine learning research* 11 (Dec): 3371–3408.

Vinodhini, G, and RM Chandrasekaran. 2012. "Sentiment analysis and opinion mining: a survey". *International Journal* 2 (6): 282–292.

Vinyals, Oriol, and Quoc Le. 2015. "A neural conversational model". *arXiv preprint arXiv:1506.05869*.

Waibel, Alexander, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. 1990. "Phoneme recognition using time-delay neural networks". In *Readings in speech recognition*, 393–404. Elsevier.

Wang, Jianglin, and Michael Johnson. 2012. "Residual phase cepstrum coefficients with application to cross-lingual speaker verification". In *Thirteenth Annual Conference of the International Speech Communication Association*.

Weizenbaum, Joseph. 1976. "Computer power and human reason: From judgment to calculation."

Wilcoxon, Frank. 1945. "Individual comparisons by ranking methods". *Biometrics bulletin* 1 (6): 80–83.

Wu, Dekai, and Pascale Fung. 2009. "Can semantic role labeling improve SMT". In *Proceedings of the 13th Annual Conference of the EAMT*, 218–225.

Wu, Zhizheng, Ali Khodabakhsh, Cenk Demiroglu, Junichi Yamagishi, Daisuke Saito, Tomoki Toda, and Simon King. 2015. "SAS: A speaker verification spoofing database containing diverse attacks". In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 4440–4444. IEEE.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. "Show, attend and tell: Neural image caption generation with visual attention". In *International Conference on Machine Learning*, 2048–2057.

Yu, Fisher, and Vladlen Koltun. 2015. "Multi-Scale Context Aggregation by Dilated Convolutions". *CoRR* abs/1511.07122. arXiv: 1511.07122.

Zeng, Wenyuan, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2016. "Efficient summarization with read-again and copy mechanism". *arXiv preprint arXiv:1611.03382*.

Zhou, Bowen, Laurent Besacier, and Yuqing Gao. 2007. "On efficient coupling of ASR and SMT for speech translation". In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, 4:IV–101. IEEE.

Zhou, Zhi-Hua, Jianxin Wu, and Wei Tang. 2002a. "Ensembling neural networks: Many could be better than all". *Artificial Intelligence* 137 (1): 239–263.

— . 2002b. "Ensembling neural networks: many could be better than all". *Artificial intelligence* 137 (1-2): 239–263.

Zhukov, Vlad, Eugene Golikov, and Maksim Kretov. 2017. "Differentiable lower bound for expected BLEU score". *arXiv preprint arXiv:1712.04708*.

Zoph, Barret, and Quoc V Le. 2016. "Neural architecture search with reinforcement learning". *arXiv preprint arXiv:1611.01578*.