Advanced Quality Measures for Speech Translation

Ngoc Tien Le

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Ngoc Tien LE

Thèse dirigée par **Laurent BESACIER**
et Co-encadrée par **Benjamin LECOUTEUX**

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)**

Advanced Quality Measures For Speech Translation

Thèse soutenue publiquement le **29 Janvier 2018**,
devant le jury composé de :

M. Frédéric BÉCHET
Professeur, Aix-Marseille Université (AMU), Président
M. Yannick ESTÈVE
Professeur, Université du Mans, Rapporteur
M. Georges LINARÈS
Professeur, Université d'Avignon et des pays de Vaucluse, Rapporteur
M. Benjamin LECOUTEUX
Maître de Conférences, Université Grenoble Alpes, Examinateur
M. Laurent BESACIER
Professeur, Université Grenoble Alpes, Directeur de thèse

“Success consists of going from failure to failure without loss of enthusiasm."

Winston Churchill

Abstract

The main aim of this thesis is to investigate the automatic quality assessment of spoken language translation (SLT), called Confidence Estimation (CE) for SLT. Due to several factors, SLT output having unsatisfactory quality might cause various issues for the target users. Therefore, it is useful to know how we are confident in the tokens of the hypothesis. Our first contribution of this thesis is a toolkit LIG-WCE which is a customizable, flexible framework and portable platform for Word-level Confidence Estimation (WCE) of SLT.

WCE for SLT is a relatively new task defined and formalized as a sequence tagging problem in which each word of SLT output is marked as one of binary labels (*good* or *bad*) in agreement with a large feature set. We propose several word confidence estimators (WCE) based on our automatic evaluation of transcription (ASR) quality, translation (MT) quality, or both (combined/joint ASR+MT). We built a corpus that contains 6.7k utterances in which each quintuplet consists of ASR hypothesis, verbatim transcript, text translation, speech translation and post-edition of translation. We performed several experiments for WCE using joint ASR and MT features to show that MT features remain the most influent while ASR features can bring interesting complementary information.

As another contribution, we propose two methods to disentangle ASR errors and MT errors, where each word in the SLT hypothesis is tagged as *good*, *asr_error* or *mt_error*. We thus explore the contributions of WCE for SLT in finding out the source of SLT errors.

Furthermore, we propose a simple extension of WER metric in order to penalize differently substitution errors according to their context using word embeddings. For instance, the proposed metric should catch near matches (mainly morphological variants) and penalize less this kind of error which has a more limited impact on translation performance. Our experiments show that the correlation of the new proposed metric with SLT performance is better than the one of WER. Oracle experiments are also conducted and show the ability of our metric to find better hypotheses (to be translated) in the ASR

N-best. Finally, we present and analyze a preliminary experiment in which ASR tuning is applied by our new metric.

To conclude, we have proposed several prominent strategies for CE of SLT that could have a positive impact on several applications for SLT. Robust quality estimators for SLT output can be applied to provide feedback to the user in computer-assisted speech-to-text scenarios or to re-score ST graphs.

Keywords: Quality estimation, Word confidence estimation (WCE), Spoken Language Translation (SLT), Joint Features, Feature Selection.

Résumé

Le travail présenté dans cette thèse vise à estimer automatiquement la qualité de la traduction de la parole (Speech Language Translation, SLT), via différentes mesures de confiance. Le système de traduction de la parole génère des séquences de mots contenant potentiellement des erreurs. Une sortie du système, avec une qualité insuffisante, peut engendrer différents problèmes pour les utilisateurs finaux. Par conséquent, il est est nécessaire d'identifier les zones d'incertitudes dans les hypothèses. Les mesures confiance consistent à générer une probabilité quantifiant le niveau de confiance associé à un mot. Cette probabilité pourra ensuite être utilisée comme seuil de décision afin de réévaluer une hypothèse. Dans le cadre de cette thèse, notre première contribution est le développement d'une boîte à outils flexible destinée à l'estimation de mesures confiance au niveau des mots issus d'un système de traduction automatique de la parole.

Dans le cadre d'un système de traduction de la parole reposant sur des modules parole/-traduction séparés, les premières erreurs sont produites au niveau des hypothèses de la reconnaissance automatique de la parole (RAP) puis se propagent au niveau de la traduction automatique (*Machine Translation* ou MT). Nous étudions ce phénomène via l'estimation de mesures de confiance (CE) au niveau des mots. Nos mesures de confiance se basent sur des modèles de champs aléatoires conditionnels (*Conditional Random Fields* ou CRF). Cette tâche, est définie et formalisée comme un problème d'étiquetage séquentiel dans lequel chaque mot, dans l'hypothèse du système SLT, est annoté comme *bon* ou *mauvais* selon un ensemble des traits. Nous proposons plusieurs outils permettant d'estimer la confiance des mots (WCE) aussi bien au niveau du système RAP qu'au niveau du système de traduction. Enfin nous proposons des mesures de confiance jointes entre système RAP et système MT. Ce travail de recherche est associé à la production d'un corpus spécifique, contenant 6700 phrases pour lesquelles un quintuplet a été fourni comme suit : (1) sortie du système RAP, (2) transcription issue du verbatim, (3) traduction manuelle, (4) traduction automatique de la parole et (5) post-édition manuelle de la traduction automatique. Nos multiples expérimentations, utilisant des traits joints entre RAP et MT pour l'estimation de qualité, ont montré que

les traits de MT demeurent les plus influents, tandis que les traits du RAP peuvent apporter des informations complémentaires.

Une autre contribution s'articule autour de deux méthodes permettant de distinguer les erreurs d'origine RAP de celles issues du système MT. Dans ces méthodes, chaque mot en sortie du système SLT, est annoté comme *bon*, *rap_erreur* ou *mt_erreur*. Nous proposons ainsi une méthode permettant d'identifier la source des erreurs au sein des systèmes de traduction automatique de la parole.

Finalement, nous proposons une nouvelle métrique, que nous avons appelée *Word Error Rate with Embeddings* (WER-E), plus adaptée à la tâche et permettant de s'appuyer plus fortement sur des aspects sémantiques. Nos expérimentations ont ainsi montré que la corrélation entre la nouvelle métrique et la qualité de la traduction automatique est plus élevée par rapport à l'utilisation d'un WER classique. Cette métrique a été exploitée pour générer de meilleures hypothèses de traduction automatique de la parole lors de la phase d'optimisation des scores d'hypothèses issues du système.

En conclusion, les stratégies proposées pour l'estimation de mesures de confiance montrent un impact positif dans plusieurs applications liées à la traduction automatique de la parole. En perspective, ces mesures de confiance robustes pourront être utilisées afin de ré-estimer des graphes de traduction de parole ou pour fournir des retours aux utilisateurs dans un contexte de traduction de la parole interactive.

Mots-clés : Estimation de la qualité, Estimation de confiance au niveau des mots, Traduction de la parole, mesures de confiance jointes, Sélection de traits.

Acknowledgements

First of all, I would like to thank my supervisors, **Laurent Besacier** and **Benjamin Lecouteux**, for providing me with extensive support, patience throughout the duration of my PhD. I feel very privileged and grateful to have had the opportunity to be supervised by **Laurent** et **Benjamin**, and I have had a very great deal to learn from their candid comments and experienced guidance.

I would also like to thank my committee members: Mr. Yannick Estève, Mr. Georges Linarès and Mr. Frédéric Béchet, for their constructive, insightful encouragement, comments and suggestions.

I express my sincere gratefulness to French volunteers who helped me in expending our speech corpora. I also thank the staffs of DOMUS (the experimental platform of Laboratory of Informatics in Grenoble) for their support.

My sincere thanks goes to the French government and the Ministry of Education and Training of Vietnam for their financial support in pursuing my doctoral study in France. I would also like to thank my supervisors for their role in securing funding support for me during the last three months of my PhD.

My time has been enriched during the PhD through interactions with members of IMAG. I have enjoyed countless lively academic (and non-academic) discussions with members of our team GETALP and other groups (past and present): *Sarah, David, Fréderic, Quang, Mateusz, Ruslan, Elina, Jérémy, Elodie, Alexis, Zied, Marwa, Yuko, Maha, Pathathai, Julian, Thao, Ky, Bao, Son, Quan, Hai, etc.* This list is by no means exhaustive and special thanks to everyone in IMAG with whom my time here has overlapped.

I would like to thank Mr. Jacques Ferrara. I am grateful he has been able to take time from his schedules to help me improve my French. Also, I appreciated his candid criticism and insightful comments.

Last, but certainly not least, I would like to thank my family for their endless support of my endeavors, for encouraging me to study abroad and to follow my dreams. Especially, the deep thank to my wife (**MAI Thi Huong**) and our children for their support and motivation all the time.

Contents

List of Figures

List of Tables

Abbreviations

NLP	**N**atural **L**anguage **P**rocessing
CE	**C**onfidence **E**stimation
QE	**Q**uality **E**stimation
WCE	**W**ord **C**onfidence **E**stimation
SLT	**S**poken **L**anguage **T**ranslation
SMT	**S**tatistical **M**achine **T**ranslation
LM	**L**anguage **M**odel
ASR	**A**utomatic **S**peech **R**ecognition
ML	**M**achine **L**earning
HMM	**H**idden **M**arkov **M**odel
GMM	**G**aussian **M**ixture **M**odel
SGMM	**S**ubspace **G**aussian **M**ixture **M**odel
UBM	**U**niversal **B**ackground **M**odel
DNN	**D**eep **N**eural **N**etwork
EM	**E**xpected **M**aximization
OOV	**O**ut-**O**f-**V**ocabulary
WER	**W**ord **E**rror **R**ate
WER-E	**W**ord **E**rror **R**ate with Embedding
TER	**T**ranslation **E**rror **R**ate
HTER	**H**uman-mediated **T**ranslation **E**rror **R**ate
BLEU	**Bi**-**L**ingual **E**valuation **U**nderstudy
METEOR	**M**etric for **E**valuation of **T**ranslation with **E**xplicit **OR**dering
POS	**P**art-**O**f-**S**peech
w/o	without

Chapter 1

Introduction

1.1 Motivation

Natural language processing (NLP) is an increasingly important area in applied linguistics. NLP can lead to many applications, such as machine translation (MT), Automatic Speech Recognition (ASR), Spoken Language Translation (SLT), Information Extraction, Summarization, etc.

However, the challenges for speech translation (such as the size of training corpus, domain mismatch, rare words, speech dis-fluencies, etc) decrease the quality of speech translation system. Therefore, we need a method to judge automatically the quality of SLT system. It is called Confidence Estimation (CE) for SLT, allowing us to know if a system produces (or not) user-acceptable outputs. Indeed, in interactive speech to speech translation, CE helps to judge if a candidate is uncertain (and ask the speaker to rephrase or repeat). For speech-to-text applications, CE may tell us if output translations are worth being corrected or if they require retranslation from scratch.

In ASR or MT, there are many approaches of CE at different levels that obtained interesting achievements such as document-level CE [Scarton and Specia, 2014] [Scarton et al., 2016], sentence-level CE [Blatz et al., 2004] [Specia et al., 2009] [Shah et al., 2016], phrase-level CE [Specia and Giménez, 2010] [Logachva and Specia, 2015] [Blain et al., 2016], word-level CE [Ueffing et al., 2003a] [Ueffing and Ney, 2005]

[Ueffing and Ney, 2007] [Bach et al., 2011] [Luong et al., 2013a] [Luong et al., 2013b] [Besacier et al., 2014] [Besacier et al., 2015] [Servan et al., 2015] [Logacheva et al., 2016] [Le et al., 2016b].

In this thesis, we focus on the word-level CE on the candidates of SLT system. We formalize it as a sequence labeling issue in which each word in SLT output is assigned by a quality score or a quality label in accordance with a large feature set. We also propose several word confidence estimators (WCE) based on our automatic evaluation of transcription (ASR) quality, translation (MT) quality, or both (combined / joint ASR+MT).

Furthermore, this thesis had the following goals:

- Inheriting the published speech corpora in [Besacier et al., 2014], we extended its size for our experimental settings and made it available to the research community.
- Studying various types of features for CE in SLT and then proposing methods to combine them
- Using our CE system for SLT to apply on various tasks such as re-ranking *N*-best list and identifying source of SLT errors
- Studying and proposing a novel automatic metric to tune ASR in a SLT context.

1.2 Main Contributions

After presenting the goals, we can emphasize on the following contributions:

1. Extending speech corpora for a French-English speech translation task that was initially presented in [Besacier et al., 2014].
2. Proposing an advanced features set for both ASR + MT systems and then building WCE system for ASR as well as WCE system for SLT based on ASR features, MT features, combined / joint ASR + MT features.

3. Exploring the usefulness of ASR and MT features in WCE system for SLT.

4. Proposing methods to disentangle ASR and MT Errors in Speech Translation.

5. Proposing an automatic metric extending word error rate (WER) that is better correlated with SLT performances.

1.3 Thesis Overview

The rest of this thesis is organized into two parts. In the first one (two first chapters), we summarize the state-of-the-art techniques for Confidence Estimation (CE) in Spoken Language Translation. In the second one (from Chapter 4 to Chapter 7), we present our contributions.

Indeed, Chapter 2 and Chapter 3 begin by laying out the theoretical dimensions of the research, and providing the concepts and the terminologies related to ASR, SMT and SLT, respectively. They also provide the descriptions about relevant general themes for CE (machine learning strategies and the metrics used to assess the CE performance). Note that Chapter 3 is concerned with the methodology used for this study. It also presents the conventional feature set and the metrics used in this thesis.

In Chapter 4, we illustrate the characteristics of the corpora used in this thesis. In addition, Chapter 4 goes over the details on how to build a robust WCE system for SLT. Then, we analyse the results of the preliminary experimentations. This chapter also expends an initial speech corpus and presents our flexible open-source (*LIG-WCE Toolkit*) used in this thesis.

In Chapter 5, we propose two methods using proposed predictor features whether to combine ASR features with MT features or to put them into joint strategies. Moreover, in this chapter, we also describe the feature selection technique that help us rank the significant indicator features (ASR features or MT features) in term of performance for CE in SLT.

In Chapter 6, we propose two methods to disentangle ASR and MT errors in speech translation by automatically detecting SLT errors' origin (is it due to ASR or to MT?)

Chapter 7 proposes a novel automatic metric to evaluate ASR candidates using word embedding.

Finally, in the conclusion section, we summarize the key contributions of the thesis as well as potential future researches.

Chapter 2

Main Concepts in Spoken Language Translation (SLT)

In this chapter, we introduce Spoken Language Translation (SLT), Automatic Speech Recognition (ASR) and automatic metrics to evaluate the output of Machine Translation (MT).

Regarding Machine Translation (MT), we review the different types of Machine Translation systems, how they are used to produce the translation hypotheses and the choice among acceptable translation candidates. More specifically, we will discuss about the components of a Statistical Machine Translation (SMT)[1].

Concerning Automatic Speech Recognition, this chapter also gives a brief overview of the methods and some terminologies used in this thesis.

This chapter has been divided into five sections. The first section (2.1) and the second section (2.2) deal with a brief definition of Machine Translation and Automatic Speech Recognition, respectively. We then present an overview of the specificities of Spoken Language Translation in Section 2.3. In addition, some useful metrics to estimate the quality of MT output are given in Section 2.4. Finally, Section 2.5 concludes this chapter.

[1]Neural Machine Translation (NMT) gained more and more attention during this PhD but we did not use it so we decided to not present it in this state-of-the-art chapter.

2.1 Machine Translation (MT)

2.1.1 Introduction

Researchers have shown an increased interest in MT since the 1950s [Hutchins, 1995]. Hutchins [2007] briefly presented the historical perspectives of MT. While a variety of definitions of the term 'Machine Translation' (MT) have been suggested, this thesis will use the definition suggested by Hutchins [1995] who saw it as computerized systems responsible for the production of translations with or without human assistance. It is also known as automated translation that is a subfield of computational linguistics.

Due to human involvement and mechanization, there are three categories of translation, such as traditional human translation, Machine-Aided Translation (MAT) and Automatic Machine Translation [Slocum, 1985]. MAT, also called Computer-Aided Translation (CAT) can be divided into two subgroups such as Human-Aided Machine Translation (HAMT) and Machine-Aided Human Translation (MAHT).

- **Human-Aided Machine Translation (HAMT)** refers to a system where the computer program generates the translation hypotheses for a given source sentence. The machine translation process benefits from human assistance when needed (for instance, asking the human translator to choose the best translation hypothesis from proposed hypotheses by machine, or asking to determine the best meaning for a target word/phrase).
- **Machine-Aided Human Translation (MAHT)** refers to a system in which the translation hypotheses are produced by the human translator. During the process, the translation process is aided by the computer (for example, an electronic bilingual terminology is provided, a pre-translation is provided to the translator, etc.).

There are several approaches to build automated MT engines, for example linguistic-based MT (direct, transfer, interlingual MT), computational-based MT (rule-based, corpus-based, example-based, statistical, neuronal MT) [Nagao, 1989] [Boitet, 2008], as discussed in the forthcoming sections.

2.1.2 Machine Translation Approaches

2.1.2.1 Different levels of transfer

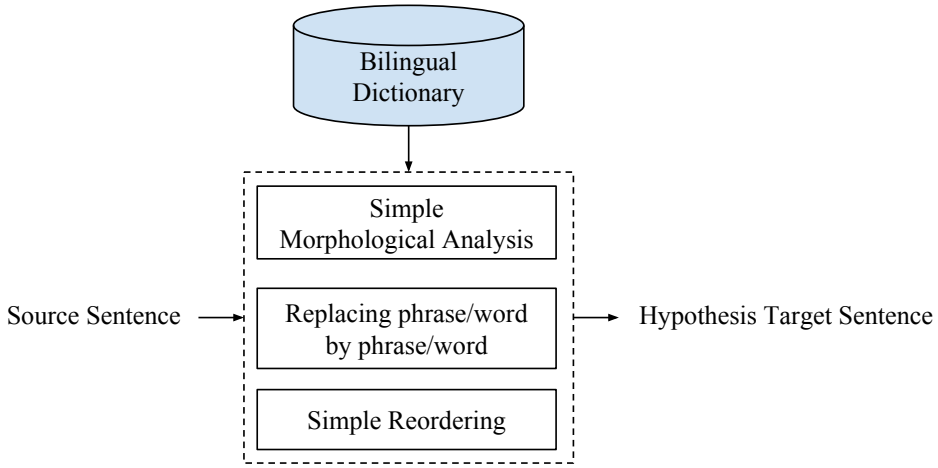

Figure 2.1: Direct Machine Translation Diagram.

- **Direct Machine Translation:** This approach was used to allow systematic, simple and fast replacing of source phrase/word by target phrase/word , as shown in Figure 2.1. Figure 2.1 shows that the direct transfer method is used at the word/phrase level. It is efficient when there is few syntactic divergence between source and target language (no need of in-depth analysis of morphology and syntax). Moreover, it is also used for small vocabulary (and specific) tasks. So, this strategy can be seen as a word-by-word translation approach with basic grammatical adjustments.

- **Transfer Machine Translation:** In Transfer-Driven approach, three modules are involved, as presented in Figure 2.2. It begins by morpho-syntactic analysis of the source sentence. It then goes on to the application of transformation rules (for instance, vocabulary and grammar rules) adjusting those to target language representations. Finally, generation in the target language is performed [Arnold and Tombe, 1987]. Note that this approach does ot take into account semantic ambiguity of source words.

- **Interlingual Machine Translation:** In this strategy, we replace the notion of "transfer" in Transfer-based MT by "interlingua". Interlingua-based MT system

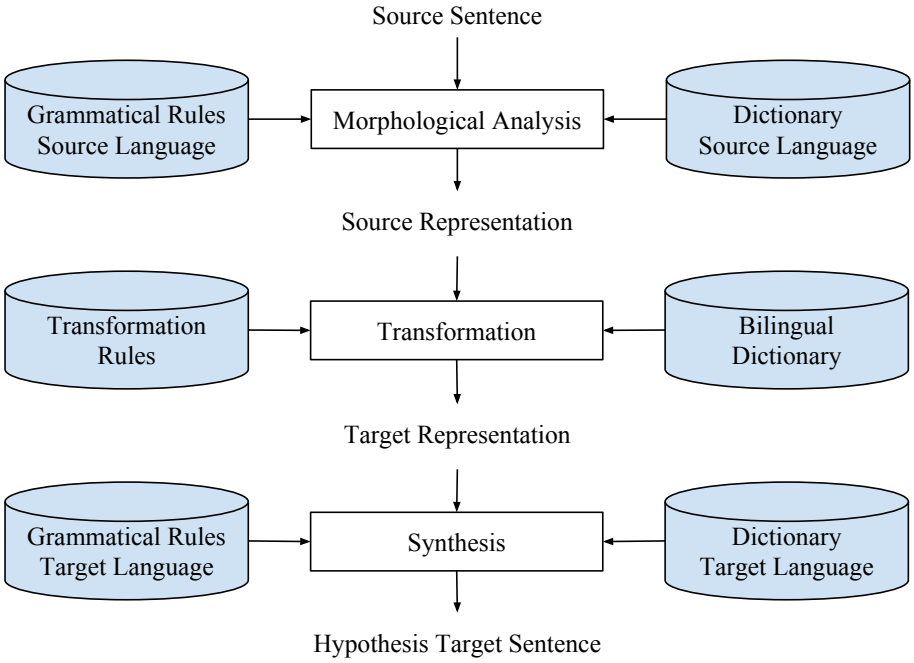

Figure 2.2: Transfer-Driven Machine Translation Diagram.

is performed in two phases, as shown in Figure 2.3. Firstly, the source sentence is analyzed into an abstract universal language-independent (interlingual) representation. Then, the target sentence is generated [Carbonell et al., 1992]. One of the most difficult problem of this method is to choose the interlingua which will contain semantic representation [Guerra, 2000]. The effectiveness of the interlingual technique has been presented in a report by Hutchins [1995].

Figure 2.4 presents the differences of above three methods. It is also called as the Vauquois Triangle [Vauquois, 1968].

As shown in Figure 2.4, there is no need of analysis and generation in the Direct MT strategy. But it uses some simple analysis rules and some rules for direct translation. The indirect strategies (Transfer-driven MT and Interlingual MT) differ in the depth of analysis.

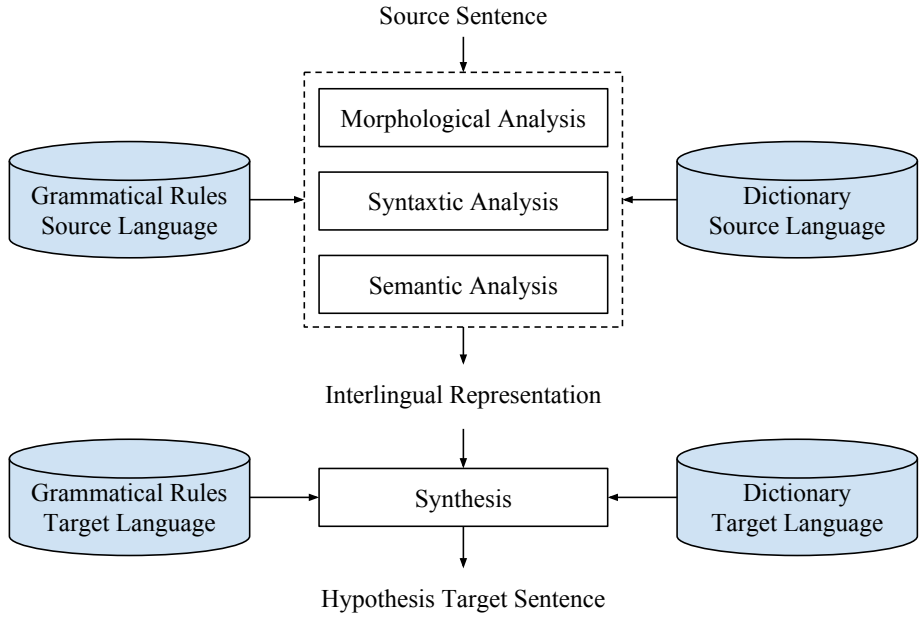

Figure 2.3: Interlingual Machine Translation Diagram.

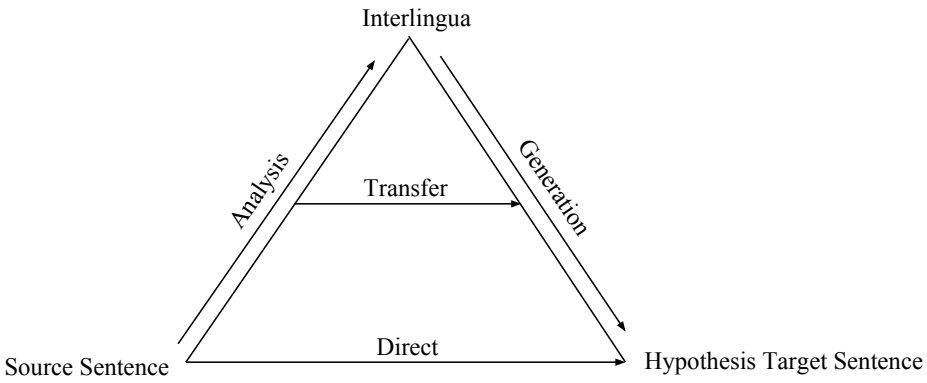

Figure 2.4: Vauquois Triangle showing the differences between Direct, Transfer and Interlingual MT strategies.

2.1.2.2 Different types of computation

- **Rule-Based Machine Translation (RBMT)** provides a conceptual theoretical framework based on built-in linguistic rules based on the morphological, syntactic and semantic analysis of both source and target sentences (see [Hutchins, 1986]). RBMT systems are conceptually indirect approaches since they use three main processes: a string-to-tree parser (analysis phase), a rule-based tree-to-tree transformer (transfer phase) and a tree-to-linear-string generator (systhesis phase). RBMT systems generate translation hypotheses with reasonable quality

if source sentence is covered by their knowledge [Carbonell et al., 2006]. However, building RBMT system is expensive and time consuming because linguistic resources need to be hand-crafted by linguistics experts. Another important practical implication is that adding new rules or updating existed rules in this system is not easy [Berwick and Fong, 1990]. So, it is hard to deal with ambiguity problems as well as idiomatic expressions [Dugast et al., 2008].

- **Statistical Approach to Machine Translation (SMT)** is an prevalent approach to MT based on statistical analysis in order to build the dictionaries and the translation rules contrasting with RBMT [Koehn, 2010]. In order to build them, bilingual parallel corpora are used. The approach is based on statistical analysis and extracts the translation probabilities of the words/phrases/syntax, etc. Weaver [1949] is the first to present translation using statistical methods and information theory. This view is extended by Brown et al. [1990] who proposed the first models for SMT, based on Bayes theory, now called *IBM models*. SMT models generate translation candidates of the source sentence and select the best one according to a maximum likelihood decision. This approach is described in more details in Section 2.1.3.

- **Example-Based Machine Translation (EBMT)** (also known as Memory-Based Translation) is also based on empirical analysis depending on the bilingual text corpora with differences in the matching and recombination phases. In addition, according to [Güvenir and Cicekli, 1998], EBMT can be defined as follows: "EBMT approach basically refers to analyzing morphological and stemming". SMT systems essentially generate statistical parameters from the bilingual corpus after preprocessing and training phases without guarantee to reproduce an observed sample. They cannot guarantee the same hypothesis output for a given source sentence, whereas EBMT systems can generate the same translation output from a given source sentence. Nagao [1984] illustrates the three main tasks of EBMT: analysis (matching the patterns against given bilingual corpus), transfer (determining the relevant translation patterns) and generation (recombining the related translation patterns into the output hypothesis). There are many useful techniques used for matching task, such as character based matching, word

based matching, annotated word based matching, structure based matching, etc [Somers, 1999].

- **Hybrid Machine Translation (HMT)** is a method borrowing from several different MT strategies, for instance rule-based and statistical techniques. The major objective of this approach is to combine the advantages of each component MT paradigm allowing to increase the accuracy of translation candidates. Costa-jussá and Fonollosa [2015], España-Bonet and Costa-jussà [2016] present an overview of current trends and applications in hybrid MT.

- **Neural Machine Translation** According to a definition provided by Luong et al. [2016], Neural Machine Translation (NMT) is "the approach of modeling the entire MT process via one big artificial neural network". Goldberg [2016] proposes good tutorial of neural network models for natural language processing. NMT systems are also known as sequence-to-sequence models or encoder-decoder networks [Kalchbrenner and Blunsom, 2013] [Sutskever et al., 2014]. In a nutshell, the encoder encodes an source sentence into a fixed (compact) representation, while the decoder generates a sequence of symbols (words or characters) given the source sentence representation as well as the previously generated symbols. Recurrent Neural Networks (RNNs) are generally used for encoder and decoder components while recent approaches have proposed to use an attention mechanism [Bahdanau et al., 2015] (somehow equivalent to the alignment model in SMT) in order to improve translation performance. Up to now, the research has tended to focus on NMT as well as on HMT. Furthermore, there are several challenges to solve for future NMT [Luong et al., 2016], including NMT with low resources which is still less efficient than SMT[2].

2.1.3 Statistical Machine Translation (SMT)

This section gives a brief overview of Statistical Machine Translation (SMT) approach. This approach is based on three main components (Language Model, Translation Model and Search Process) and it is derived from the analysis of bilingual text corpora.

[2]see for instance https://duyvuleo.github.io/ws17mt/

2.1.3.1 Language Modeling

One of the standard Language Modeling model used in Machine Translation is the N-gram model. This model represents the probability of generating the word at position n given the previous $n-1$ words, so-called the history.

Using the chain rule of probability [Jurafsky and Martin, 2000], we could have the probabiliy of a sentence $P(W)$ where $W = (w_1, w_2, w_3, ..., w_n) = w_1^n$, w_i is the i^{th} word in sentence W, $1 \leq i \leq n$,

$$\begin{aligned} P(W) = P(w_1^n) &= P(w_1, w_2, w_3, ..., w_n) \\ &= P(w_1) P(w_2|w_1) P\left(w_3|w_1^2\right) ... P\left(w_n|w_1^{n-1}\right) \\ &= \prod_{i=1}^{n} P\left(w_i|w_1^{i-1}\right) \end{aligned} \tag{2.1}$$

The above chain rule is used to measure the probability of a sentence and estimate the conditional probability of word at position i^{th} with given all of previous words. However, it is not possible to calculate the probability of a word with given a sequence of previous words, $P\left(w_n|w_1^{n-1}\right)$. Thus, we use Markov assumption to assess the probability of a word depending only on a short history (2-gram to 5-gram). For example, in the bigram assumption, the probability of each word in Equation 2.1 is defined as,

$$P(W) = P(w_1^n) \approx \prod_{i=1}^{n} P(w_i|w_{i-1}) \tag{2.2}$$

where a particular bigram probability $P(w_i|w_{i-1})$ is computed as,

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \tag{2.3}$$

where C is a *count* function. Therefore, using the N-gram assumption for the probability of each word, we have the equation:

$$\begin{aligned} P(W) = P(w_1^n) &= P(w_1, w_2, w_3, ..., w_n) \\ &= P(w_1) P(w_2|w_1) P\left(w_3|w_1^2\right) ... P\left(w_n|w_1^{n-1}\right) \\ &\approx P(w_1) P(w_2|w_1) ... P\left(w_{N-1}|w_1^{N-2}\right) \prod_{i=N}^{n} P\left(w_i|w_{i-N+1}^{N-1}\right) \end{aligned} \tag{2.4}$$

Language model (LM) is usually generated from a given set of words that are called the in-vocabulary words. The other words are considered as Out-Of-Vocabulary (OOV)

words. One problem is that the LM may give a probability 0 to OOV words. To avoid this issue, a variety of smoothing methods allow to generate probabilities for unseen tokens such as Additive smoothing, Good-Turing estimate [Good, 1953], Jelinek-Mercer smoothing (interpolation) [Jelinek and Mercer, 1980], Katz smoothing (backoff) [Katz, 1987], Witten-Bell smoothing [Witten and Bell, 1991], Kneser-Ney smoothing [Kneser and Ney, 1995]. A summary of the main findings and of the principal issues is provided in [Chen and Goodman, 1996], [MacCartney, 2005] and [Koehn, 2010].

2.1.3.2 Translation Modeling

a. Word-based Translation Modeling

This subsection focuses on word-based translation and in particular on IBM translation models.

In general, statistical translation models are based on the concept of word alignments from bilingual corpora during training phase. The following notations are needed to mathematically define word-based TM:

- $f^k \in \left\{f^{(1)}, f^{(2)}, ..., f^{(n)}\right\}$ is the k^{th} sentence in source corpus having n sentences.
- $e^k \in \left\{e^{(1)}, e^{(2)}, ..., e^{(n)}\right\}$ is the k^{th} sentence in target corpus having n sentences and is aligned to f^k.
- $a^k \in \left\{a^{(1)}, a^{(2)}, ..., a^{(n)}\right\}$ is the k^{th} word alignment set between f^k and e^k.
- $f = (f_1, f_2, ..., f_m)$, where m is the length of the source sentence and $f_i, i \in \{1, 2, ..., m\}$ is the i^{th} word in the source sentence f.
- $e = (e_1, e_2, ..., e_l)$, where l is the length of the target sentence and $e_j, j \in \{1, 2, ..., l\}$ is the j^{th} word in the target sentence f.
- $a = (a_1, a_2, ..., a_m)$, where $a_i, i \in \{1, 2, ..., m\}$ is one alignment information for source word f_i, each alignment information take any value from 0 to l. For example, if there is alignment between f_i and e_j, $a_i = j$. Especially, we define that e_0

is a special target word, also called *NULL*. In other words, $a_i = 0$ if source word f_i is aligned to *NULL* word.

- $\mathscr{F}$, $\mathscr{E}$ are a finite set of source words and target words, respectively.
- M, L are the maximum lengths of source and target sentences, respectively.
- $p(f|e), f_i \in \mathscr{F}, e_j \in \mathscr{E} \cup \{NULL\}$ is the conditional probability for translating from source sentence f to candidate sentence e.
- $q(j|i,l,m)$ is the probability of alignment $a_i = j$ with given the length of source sentence m and target sentence l, where $l \in \{1,2,...,L\}$, $m \in \{1,2,...,M\}$, $i \in \{1,2,...,m\}$, $j \in \{1,2,...,l\}$.

We now turn to the problem of modeling the conditional probability $p(f|e)$:

$$p(f|e) = p(f_1...f_m|e_1...e_l) = p(f_1...f_m|e_1...e_l,m) \tag{2.5}$$

This probability can be estimated with increasingly complex models that take into account lexical translation, word-reordering and word fertility that are presented in five IBM models and trained using Expectation Maximization (EM) algorithm [Dempster et al., 1977] or hidden Markov Model (HMM) algorithm [Rabiner, 1990].

*** Model 1 - Lexical Translation Probability**

IBM Model 1 is estimated:

$$q(j|i,l,m) = \frac{1}{l+1} \tag{2.6}$$

Note that, it is possible that $j = 0$, when the source word f_i is aligned to $e_0 = NULL$.

Thus,

$$p(a|e) = \frac{1}{(l+1)^m} \tag{2.7}$$

In addition, the probability of target sentence is computed:

$$p(f|e,a) = \prod_{i=1}^{m} t(f_i|e_{a_i}) \tag{2.8}$$

Therefore, Equation 2.5 is presented as follows,

$$p(f|e) = \sum_{a} p(f,a|e) = \sum_{a} p(f|e,a) * p(a|e) = \sum_{a} \frac{1}{(l+1)^m} \prod_{i=1}^{m} t(f_i|e_{a_i}) \tag{2.9}$$

Using this model to find the best word alignment between source sentence f and target sentence e, the best alignment is found:

$$\hat{a} = \underset{a}{\operatorname{argmax}}\, p(f,a|e) = \underset{a}{\operatorname{argmax}} \frac{1}{(l+1)^m} \prod_{i=1}^{m} t(f_i|e_{a_i}) = \underset{a}{\operatorname{argmax}} \prod_{i=1}^{m} t(f_i|e_{a_i}) \tag{2.10}$$

where $i \in \{1,2,...,m\}$.

*** Model 2 - Addditional Distorsion Model Probability**

As presented in IBM model 1, the alignment probability distribution is not used. IBM model 2 addresses this problem as follows,

$$\begin{aligned} p(f,a|e) &= \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} ... \sum_{a_m=0}^{l} p\left(f_1...f_m, a_1...a_m | e_1...e_l\right) \\ &= \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} ... \sum_{a_m=0}^{l} \prod_{i=1}^{m} q(a_i|i,l,m) t(f_i|e_{a_i}) \end{aligned} \tag{2.11}$$

*** Model 3 - Fertility Probability**

This model focuses on the "fertility" problem. It means that one source word can be translated into a specific number of candidate words, for example 0, 1 or more [Koehn, 2010].

*** Model 4 - Additional Relative Alignment Probability**

This model integrates a finer distorsion model which is not described here. The mathematical equations of this model are presented in [Brown et al., 1993] and [Koehn, 2010].

*** Model 5 - Fixing Deficiency Problem**

In IBM model 3,4, the position of source word generated by target word *NULL* is not modeled. Thus, IBM model 5 addresses this issue [Brown et al., 1993].

b. Phrase-based Translation Modeling

To reduce the limitations of word-based models, another approach is used for modeling TM, called phrased-based translation modeling. In this strategy, $p(f|e)$ is defined as follows [Koehn et al., 2003]:

$$p(f|e) = \prod_i \varphi\left(f_i^{'}|e_i^{'}\right) d(a_i, b_{i-1}) \tag{2.12}$$

where

- $\varphi\left(f_i^{'}|e_i^{'}\right)$ denotes the translation propability for producing the i^{th} hypothesis phrase (word sequence) $e_i^{'}$ with given i^{th} phrase $f_i^{'}$ of source sentence f.
- $d(a_i, b_{i-1})$ denotes the relative distribution probability of distortion, a_i and b_{i-1} are the begin position of source phrase translated into i^{th} hypothesis phrase and the end position of source phrase translated into $(i-1)^{th}$ hypothesis phrase, respectively.

2.1.3.3 Search Process

The objectives of search are to determine the most probable candidate in target language with given source sentence. The algorithms solving this task are often Greedy Hill-Climbing Decoding, A* Search, Beam Seach [Koehn, 2010].

```
1: place empty hypothesis into stack 0
2: for all stacks 0...n−1 do
3:   for all hypotheses in stack do
4:     for all translation options do
5:       if applicable then
6:         create new hypothesis
7:         place in stack
8:         recombine with existing hypothesis if possible
9:         prune stack if too big
10:      end if
11:    end for
12:  end for
13: end for
```

Figure 2.5: Pseudo-code for the stack decoding heuristic (taken from [Koehn, 2010]).

Beam search algorithm is briefly presented in Figure 2.5. It uses stacks in order to contain the possible hypotheses and each stack holds only a beam of one candidate. In addition, the hypothesis sentence is generated from left to right within the partial translation.

2.2 Automatic Speech Recognition (ASR)

2.2.1 Introduction

The purpose of this section is to review the literature on Automatic Speech Recognition (ASR). It begins by the brief introduction of the methods and terminologies (in ASR) used in this thesis.

Speech Recognition (also known as Automatic Speech Recognition - ASR) is the ability of a system to transcribe a speech signal input into a textual representation corresponding to the spoken word sequence.

2.2.2 ASR Architecture

General architecture of the state-of-the-art Statistical Speech Recognition system (hidden Markov model-based) is presented in Figure 2.6 and is based on 4 main modules: acoustic model, lexical model, language model and decoding algorithm.

The following sections will discuss these components in more detail.

2.2.3 Decoder

Given the observation sequence X extracted from a speech signal, in order to find the best sequence of words $\widehat{W}$, the ASR problem is defined as:

$$\widehat{W} = \underset{W}{\operatorname{argmax}} \{P(W|X)\} \tag{2.13}$$

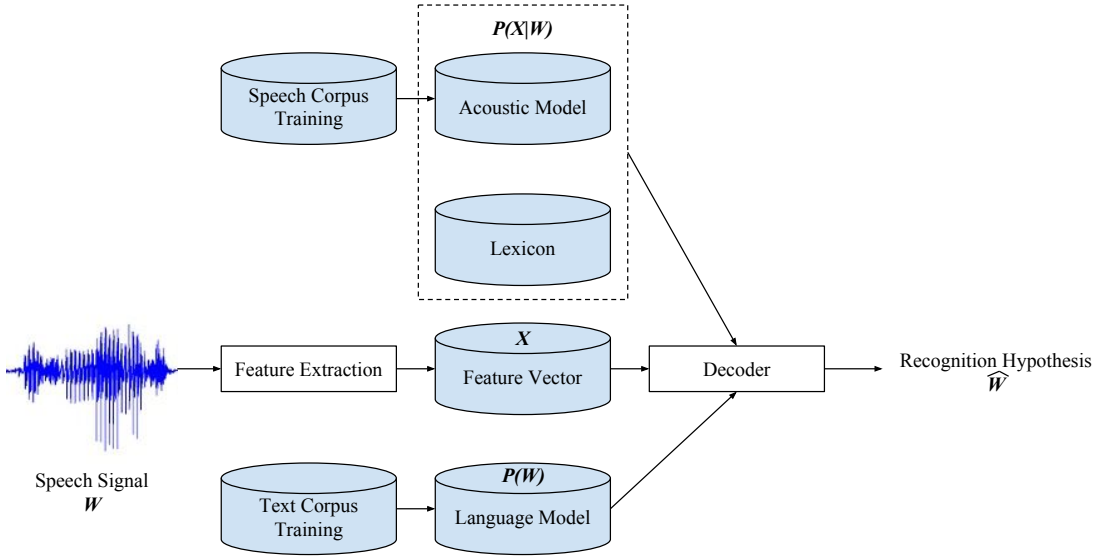

Figure 2.6: Statistical Speech Recognition state-of-the-art architecture integrating the three main components.

After using Bayes' rule, equation becomes,

$$\begin{aligned} \widehat{W} &= \underset{W}{\operatorname{argmax}} \left\{ \frac{P(X|W)\,P(W)}{P(X)} \right\} \\ &= \underset{W}{\operatorname{argmax}} \{P(X|W)\,P(W)\} \end{aligned} \tag{2.14}$$

where the term $P(X)$ is ignored since it is a constant across the various word sequences W.

In Equation 2.14, the first term $P(X|W)$, also called the likelihood of the data, is determined by an *acoustic model* and *lexical model*. And its second term $P(W)$ is typically modeled by a *language model*.

2.2.4 Feature Extraction

During feature extraction phase, speech waveforms are transformed into the observation vectors used to train the hidden Markov models [Rabiner, 1990]. The techniques used in this phase are Mel-Frequency Cepstral Coefficients (MFCCs) [Davis and Mermelstein, 1980] and Perceptual Linear Prediction (PLP) [Hermansky, 1990]. Another type of spectral analysis often used is Linear Predictive Coding (LPC) [Rabiner, 1990].

2.2.5 Language Modeling

As presented in Subsection 2.1.3.1, we often use trigram method in Language Modeling of Speech Recognition system. It is formulated as,

$$P(W) = P(w_1^n) \approx P(w_1)P(w_2|w_1)\prod_{i=3}^{n} P\left(w_i|w_{i-2}^2\right) \tag{2.15}$$

$$P\left(w_i|w_{i-2}^2\right) = P(w_i|w_{i-2},w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})} \tag{2.16}$$

More recently, Recurrent Neural Networks (RNNs) for language modeling were introduced by [Elman, 1990] and are now state-of-the-art [Mikolov et al., 2010] [Jalalvand et al., 2016]. However, we do not detail them since these RNNs were not used in our thesis.

2.2.6 Acoustic Modeling

2.2.6.1 Hidden Markov Model (HMM)

Note that each speech waveform W in Equation 2.14 is determined by a sequence of frames generated from a sequence of hidden states S represented by a subword/phonetic segmentation.

Therefore, Equation 2.14 becomes,

$$\widehat{W} = \underset{W}{\operatorname{argmax}}\{P(W)P(X,S|W)\} \tag{2.17}$$

where the term $P(X,S|W)$ is in statistical speech system that can be modeled by hidden Markov model. It is illustrated in Figure 2.7.

In order to define an HMM completely [Rabiner, 1990], we have the following notations:

$S = \{s_1, s_2, ..., s_N\}$: a set of N hidden states of the model

$\Pi_0 = \{\pi_i = P(s_i = 0)\}$: a set of transition probabilities of the initial states, where $p(s_i = 0)$ denotes the probability of the i^{th} state (s_i) at initial time $(t = 0)$, $\sum_{i=1}^{N} \pi_i = 1$

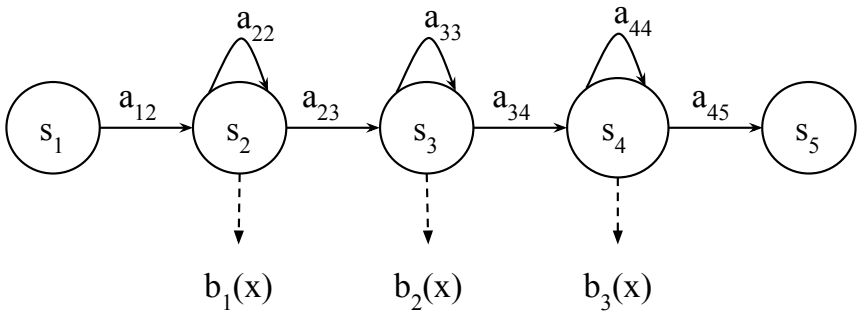

Figure 2.7: A three-state left-to-right HMM model.

$X = \{x_1, x_2, ..., x_K\}$: a set of K observation symbols

$A = \{a_{ij}\}$, $1 \leq i, j \leq N$: a set of state transition probabilities, where $a_{ij} = P(s_j = t | s_i = t-1)$ denotes the transition probability from state i to state j. It is noted that $a_{ij} \geq 0$ and $\sum_{j=1}^{N} a_{ij} = 1, 1 \leq i \leq N$

$B = \{b_j(k)\}, 1 \leq k \leq K, 1 \leq j \leq N$: a set of the probabilistic distribution in each of the states, where $b_j(k) = p(x_k | s_j = t)$ denotes the probability of the k^{th} observation (x_k) from a state s_j at time t. It is noted that $b_j(k) \geq 0$ and $\sum_{k=1}^{K} b_j(k) = 1, 1 \leq j \leq N$

Therefore, we can build the following compact notation to denote an HMM with discrete probability distributions:

$$\Lambda = (A, B, \Pi) \tag{2.18}$$

Given the above definition of HMM, there are three fundamental problems of interest:

1. Evaluation: the parameters of the HMM acoustic model A, B, Π are measured by the Forward-Backward methods [Rabiner and Juang, 1993]. Hence, it may be used to calculate the probability of the observation set given the model Λ, $p(X|\Lambda)$.

2. Decoding: Viterbi algorithm [Viterbi, 1967] [Forney, 1973] can be used to find the one-best state sequence (path) given observation set. More detailed information can be found in [Rabiner and Juang, 1993].

3. Learning: an alternative method for modifying the parameters of HMM A, B that maximizes the probability of X is by using Baum-Welch algorithm (Expectation-Maximization (EM) method [Dempster et al., 1977]) or using gradient approaches [Levinson et al., 1983]:

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}}\, p(X|\Lambda) \tag{2.19}$$

Using the above theory and the Markov assumption for Equation 2.20, we have:

$$\begin{aligned} \widehat{W} &= \underset{W}{\operatorname{argmax}} \left\{ P(W) \sum_{S} P(X|S) P(S|W) \right\} \\ &= \underset{W}{\operatorname{argmax}} \left\{ P(W) \sum_{S} \prod_{t} P(x_t|s_t) P(S|W) \right\} \end{aligned} \tag{2.20}$$

where x_t, s_t denote the observation and hidden state at time t, respectively. The term $P(x_t|s_t)$ is determined by the acoustic model while the term $P(S|W)$ is estimated by the lexicon model that provides a probability of a mapping between words and subwords/-phonemes.

Moreover, Gaussian Mixture Models, Subspace Gaussian Mixture Models and Deep Neural Networks can be used to compute the probability density B in Equation 2.18.

The following sections describe in more details how this probability density function B can be estimated.

2.2.6.2 Gaussian Mixture Models (GMM)

A Gaussian Mixture Model is represented by a weighted sum of M Gaussian components, defined by the equation,

$$P(x|s) = \sum_{i=1}^{M} w_i\, \theta(x|\mu_i, \Sigma_i) \tag{2.21}$$

where x denotes a D-dimensional continuous measurements or features vector, w_i are the mixture weights and their total value is 1, $\theta(x|\mu_i, \Sigma_i)$ denotes the Gaussian component

densities and each of them is a D-variate Gaussian function as given by the equation,

$$\theta\left(x|\mu_i,\Sigma_i\right) = \frac{1}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}}\exp\left[-\frac{1}{2}\left(x-\mu_i\right)^T\sum_i^{-1}x-\mu_i\right] \tag{2.22}$$

And the notation presents the complete GMM from the mean vectors μ_i, covariance matrices Σ_i and mixture weights from all of the component densities, where $1 \leq i \leq M$,

$$\Lambda = \{w_i,\mu_i,\Sigma_i\} \tag{2.23}$$

Also, the maximum likelihood estimations of the parameters of GMMs are obtained from training data using the conventional expectation-maximization (EM) algorithm [Dempster et al., 1977] from the well-trained prior-model.

2.2.6.3 Subspace Gaussian Mixture Models (SGMM)

Subspace Gaussian Mixture Model (SGMM) is an acoustic modeling based on reducing the dimension of vector (also called subspace) that contains parameters of the mixture weights and means of a shared Gaussian mixture model [Povey et al., 2011a]. The probability model $P\left(x|s\right)$, mean μ_{ji} and mixture weights w_{ji} for each state s of a HMM can be expressed by the following equations,

$$P\left(x|s\right) = \sum_{i=1}^{I} w_{ji}\theta\left(x|\mu_{ji},\Sigma_i\right) \tag{2.24}$$

$$\mu_{ji} = \text{Mean}_i v_j \tag{2.25}$$

$$w_{ji} = \frac{\exp\left(w_i^T v_j\right)}{\sum_{i'=1}^{I}\exp\left(w_{i'}^T v_j\right)} \tag{2.26}$$

where i is the index of component Gaussian, I is the number of Gaussians for each state or substate, Mean_i denotes the mean of projection matrix of the i^{th} component Gaussian, v_j is the distinct state and $v_j \in \mathbb{R}^S$ (S is the given phonetic subspace dimension), w_i is the weight projection vector.

Using the substates, the above equations could be extended as follows,

$$P(x|s) = \sum_{m=1}^{M_j} c_{jm} \sum_{i=1}^{I} w_{jmi}\theta\left(x|\mu_{jmi},\Sigma_i\right) \quad (2.27)$$

$$\mu_{jmi} = \text{Mean}_i v_{jm} \quad (2.28)$$

$$w_{jmi} = \frac{\exp\left(w_i^T v_{jm}\right)}{\sum_{i'=1}^{I} \exp\left(w_{i'}^T v_{jm}\right)} \quad (2.29)$$

where m is the index of substate, M_j is the number of substates of state s, v_{jm} is the specific substate in the subspace vector and $v_{jm} \in \mathbb{R}^S$ and the substate weights should satisfy the next constraint,

$$\sum_{m=1}^{M_j} c_{jm} = 1 \quad (2.30)$$

SGMMs have shown better performance than conventional GMM-based in various speech recognition tasks [Lu et al., 2011].

2.2.6.4 Deep Neural Networks (DNN)

Conventional ASR systems have used GMM or SGMM based on HMM (abbreviated by HMM/GMM or HMM/SGMM) to produce the sequential structure of speech signals. However, the above models suffer from several major drawbacks in representing complex, non-linear relationships between the acoustic features and generation input of speech. Seide et al. [2011], Dahl et al. [2012], Hinton et al. [2012] showed that using DNN for acoustic modeling improves the performance of ASR systems. In this approach (HMM/DNN), GMM or SGMM are replaced by DNN for assessing and fitting between the frame of acoustic observations and each HMM state.

DNN architecture is a conventional feed-forward artificial neural network, also called Multi-Layer Perceptron (MLP) [Rosenblatt, 1962] with many hidden layers. The HMM/DNN architecture for large-vocabulary speech recognition having a L-hidden-layer DNN is illustrated in Figure 2.8. While HMM represents the sequential features of speech signal, DNN models the observations likelihood of all the senones (tied tri-phone states directly) [Dahl et al., 2012].

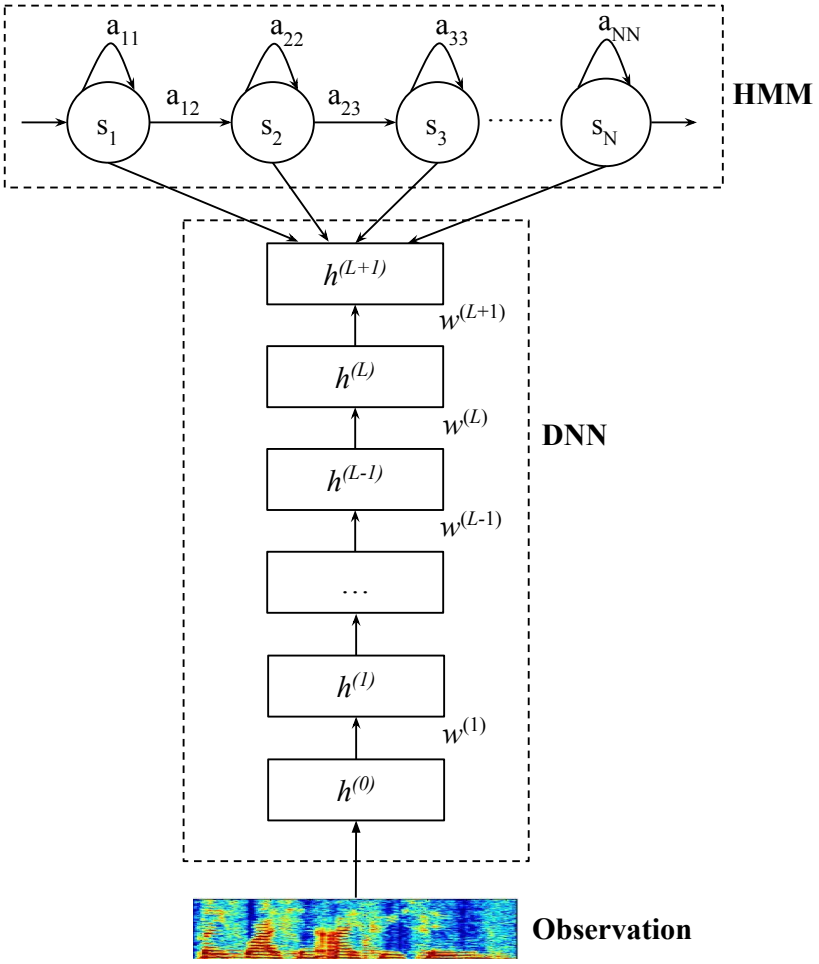

Figure 2.8: HMM/DNN architecture having L-hidden-layer DNN for large-vocabulary speech recognition .

Mathematically, each unit (hidden state) j^{th} in hidden layer l^{th} could be determined as

$$h_j^{(l)} = g\left(b_j^{(l)} + \sum_i h_i^{(l-1)} w_{ij}^{(l)}\right) \tag{2.31}$$

where $1 \leq l \leq L+1$, $g(.)$ denotes a sigmoid function, $g(a) = (1+\exp(-a))^{-1}$ (applied element-wise on vector a), $b_j^{(l)}$ denotes the bias of the j^{th} hidden unit in layer l^{th}, $w_{ij}^{(l)}$ is the weight of the relation between $h_j^{(l-1)}$ and $h_j^{(l)}$. Note that, $h^{(0)}$, also called "input layer" stands for input features. And, the last layer, $h^{(L+1)}$, also called "output layer" typically uses a softmax function for multi-class classification task or it uses a linear activation function for regression task.

Each unit in each hidden layer is typically assigned to the weighted sum of its inputs from the previous layer to a deterministic value using highly non-linear and varying

functions such as a sigmoid function, tanh, etc. Deep networks can be trained with gradient descent method. This method is sensitive to the initialisation data and back-propagation technique is often trapped in poor local minima [Hochreiter et al., 2001]. Moreover, DNN using supervised training could cause overfitting [Ling et al., 2015]. To avoid this problem, Hinton [2010] presented an unsupervised pre-training strategy by stacking Restricted Boltzmann Machine (RBM) model and then fine-tuning with back-propagation method. This unsupervised approach is also named as Deep Belief Network (DBN) [Hinton et al., 2006].

Furthermore, in HMM/DNN, the probability model to estimate the observation probabilities could be determined as:

$$P(x_t|s_t) = \frac{P(s_t|x_t)P(x_t)}{P(s_t)} \tag{2.32}$$

where $P(s_t|x_t)$ denotes the hidden state (senone) posterior probability computed from DNN, $P(x_t)$ is an constant and thus could be ignored, $P(s_t)$ is prior probability of each state computed from training corpora.

2.3 Specificities of Speech Translation

According to Tree [1995], speech disfluencies can be defined as follows: *"phenomena that interrupt the flow of speech and do not add propositional content to an utterance"*. In addition to the combination of sound and tone, there are other language auxiliaries such as gestures of the speaker. Moreover, words in spoken language are used in several ways: slang, local words, idiom, etc.

Schachter et al. [1991], Rao et al. [2007] and Segal et al. [2015] discussed about the undesired impact of disfluencies on Machine Translation and about the impacts of the errors of ASR on the performance of MT system. For instance, the speaker repeats syllables in the sentence or the speaker adds some non-lexical utterances such as "huh", "uh", "um", etc.

There are also consistency issues between general expected written inputs for translation (MT) and produced outputs by speech transcription (ASR).

Also, in continuous speech, there is no word boundary information. Thus, output of automatic transcription will generate the word sequences without punctuation marks, case, special characters, compound words, digit-numbers, etc. Thus, when translating speech from source language to target language, it is important to recover, at least partially, the above informations (or format) since MT systems are trained on (well-formed) texts.

In the scope of this thesis, we focus on the following consistencies in corpora:

- We should be able to transform back-and-forth every natural number (cardinal number), ordinal number, Roman numerals into their letter version. For example, “10” $\leftrightarrow$ “dix”; “10e” $\leftrightarrow$ “dixième”; “X” $\leftrightarrow$ “dixième”
- We should be able to add/remove (back-and-forth) punctuation (ASR should be evaluated without punctuation, while MT is taking advantage of - and should be evaluated with - punctuation). For example, “les chirurgiens de los angeles ont dit qu’ils étaient outrés, a déclaré camus.” $\leftrightarrow$ “les chirurgiens de los angeles ont dit qu’ils étaient outrés a déclaré camus”
- We should be able to add/remove (back-and-forth) abbreviations and special characters. For example, “m camus” $\leftrightarrow$ “monsieur camus”; “mme piegza” $\leftrightarrow$ “madame piegza”; “%” $\leftrightarrow$ “pourcent”; “€” $\leftrightarrow$ “euro”

2.4 Evaluation

In this section, we focus on the metrics used to measure the of language models (LM), transcription (ASR) and translation (MT).

2.4.1 Language Model Performance (Perplexity)

One metric for estimating LMs is **perplexity** that is computed on heldout dataset [Rosenfeld, 2000], defined as,

$$\textbf{perplexity}\,(W) = 2^{-\frac{1}{M}\sum_{i=1}^{N}\log_2 P(W_i)} \tag{2.33}$$

where M and N is the number of words and the number of sentences in heldout dataset, respectively, $P(W_i)$ is the probability of the i^{th} sentence evaluated by the model.

2.4.2 Transcription Performance - Word Error Rate (WER)

One of the most used quality estimation for speech recognition is Word Error Rate (WER) measurement methodology, defined as following:

$$\text{WER} = \frac{\#Ins + \#Sub + \#Del}{\#Total\ of\ Tokens\ in\ the\ Reference} \times 100\% \tag{2.34}$$

where *#Ins* denotes the number of aligned tokens that are added in the ASR hypothesis, *#Sub* is the number of words in reference that are replaced by the aligned words in the hypothesis, *#Del* is the number of words that are missed out from the reference.

Let *#Corr* be the number of words that appear on both hypothesis sentence and associated reference sentence, then WER can be also defined as:

$$\text{WER} = \frac{\#Ins + \#Sub + \#Del}{\#Corr + \#Sub + \#Del} \times 100\% \tag{2.35}$$

Note that WER could be larger than 100% when $\#Ins > \#Corr$.

2.4.3 Translation Performance

2.4.3.1 Bilingual Evaluation Understudy (BLEU)

Bilingual Evaluation Understudy (BLEU) is a method to assess the quality of a machine translation hypothesis. Papineni et al. [2002] proposed this metric and presented some

correct correlation with human evaluation of MT. However, this controversial feature was more discussed later on [Callison-Burch et al., 2006].

We define $p_1, p_2, p_3, ...p_N$ as (respectively) unigram precision (proportion of correct words among all candidate words), bigram precision (proportion of correct bigrams among all candidate bigrams), trigram precision (proportion of correct trigrams among all candidate trigrams), etc. Then we put all precisions together by computing the *geometric mean* of the given ratios as following:

$$\begin{aligned} G_{mean} &= \left(\prod_{i=1}^{N} p_i\right)^{1/N} \\ &= \exp\left(\log_e\left(\left(\prod_{i=1}^{N} p_i\right)^{1/N}\right)\right) \\ &= \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log_e(p_i)\right) \end{aligned} \tag{2.36}$$

Next, we introduce the brevity penalty (BP) that computes from the reference length r and from the hypothesis translation length c as:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1-\frac{r}{c}\right) & \text{otherwise} \end{cases} \tag{2.37}$$

Finally, BLEU is defined as following:

$$\text{BLEU} = \text{BP} \times \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log_e(p_i)\right) \tag{2.38}$$

2.4.3.2 Translation Error Rate (TER)

Translation Error Rate (TER) [Snover et al., 2006] is an automated metric that counts the number of editing operations (substitution (*#SUB*), deletion (*#DEL*), insertion (*#INS*) of a word, as well as shifts (*#SHF*) of a word or of adjacent words, needed to transform a MT hypothesis into a reference translation. TER is given by the next equation:

$$\text{TER} = \frac{\text{number of editing operations}}{\text{average length of references}} = \frac{\#SUB + \#DEL + \#INS + \#SHF}{\text{average length of references}} \tag{2.39}$$

2.4.3.3 Human-mediated Translation Error Rate (HTER)

Human-mediated Translation Error Rate (HTER) [Snover et al., 2006] is a semi-automatic (edit-distance) metric that is also used to estimate the quality of MT system. It depends on skilled monolingual human editors that correct MT hypotheses in order to convey the original meaning of the source sentence. HTER can be seen as TER where references have been generated by humans who post-edited the MT output itself.

2.4.3.4 Metric for Evaluation of Translation with Explicit ORdering (METEOR)

Metric for Evaluation of Translation with Explicit ORdering (METEOR) [Banerjee and Lavie, 2005] was proposed to better correlate with human judgements by using more than word-to-word alignments between a hypothesis and some references. The alignment is made according to three modules: the first stage uses exact match between word surface forms, the second one compares word stems and the third one uses synonyms from a lexical resource such as WordNet.

Mathematically, METEOR score uses $F_{measure}$[3] and an additional *Penalty* factor as follows:

$$F\text{-}measure = \frac{precision \times recall}{\alpha \times precision + (1-\alpha) \times recall} \tag{2.40}$$

$$Penalty = \gamma \times \left(\frac{\text{number of chunks}}{\text{number of matches}} \right)^{\beta} \tag{2.41}$$

$$METEOR\ score = (1 - Penalty) \times F\text{-}measure \tag{2.42}$$

2.5 Conclusion

In this chapter, we described the basic concepts of machine translation (MT) and automatic speech recognition (ASR), as well as the specificities of spoken language translation (SLT). Finally, metrics used to assess ASR, MT and SLT performance were presented.

[3] *precision*, *recall* and *F-measure* will be discussed in more detail in Subsection 3.6.1.

In the next chapter, we will review the specific approaches used in Confidence Estimation (CE) which is the main subject of this thesis.

Chapter 3

Main Concepts in Confidence Estimation (CE)

This chapter provides the background on Confidence Estimation (CE) in Spoken Language Translation (SLT), Machine Learning (ML) techniques and automatic metrics used in Confidence Estimation. First, existing approaches for various levels of CE in SLT are summarized. Second, we focus more on Word level Confidence Estimation system for SLT. We also present the WCE methods and performance metrics.

The overall structure of this chapter takes the form of seven sections. This chapter begins by a brief overview of automatic quality assessment of spoken language translation in Section 3.1. Section 3.2 explains that CE can be estimated at several levels to predict the quality of speech translation output. Then, we formalize the problem in Section 3.3. Section 3.4 presents the set of features used for WCE in SLT. Whereas Section 3.5 presents varying techniques to train / to label / to optimize the performance of CE system, in Section 3.6 we explain how to evaluate the performance of CE system. Finally, a conclusion gives a brief summary of this chapter.

3.1 Introduction

Automatic quality assessment of spoken language translation (SLT), also named confidence estimation (CE) or quality estimation (QE), is an important topic because it allows to know if a system produces (or not) user-acceptable outputs. In interactive speech to speech translation, CE helps to judge if a translated turn is uncertain. For speech-to-text applications, CE may tell us if output translations are worth being corrected or if they require new translation from scratch. Moreover, an accurate CE can also help to improve SLT itself through a second-pass N-best list re-ranking or search graph re-decoding [Bach et al., 2011] [Luong et al., 2014a] [Besacier et al., 2015]. Consequently, building a method which is able to point out the correct parts as well as detect the errors in a speech translated output is crucial to tackle above issues. Basing on the use-cases, we use CE in several levels such as document-level CE, sentence-level CE, phrase-level CE, word-level CE which will be presented in the next section.

Indeed, the first works about confidence estimation for MT [Ueffing et al., 2003b] [Blatz et al., 2004] were inspired by work done in automatic speech recognition [Wessel et al., 2001]. The combination of internal and external features was used in these systems. Later on, Xiong et al. [2010] integrated POS tagging and other external features. In the same way, Felice and Specia [2012] proposed 70 linguistic features for quality estimation at sentence level.

Recent workshops proposed some shared evaluation tasks of WCE systems, in which several attempts of participants to mix internal and external features were successful. The estimation of the confidence score uses mainly classifiers like Conditional Random Fields [Han et al., 2013] [Luong et al., 2014b], Support Vector Machines [Langlois et al., 2012] or Perceptron [Bicici, 2013].

Further, some investigations were conducted to determine which feature seems to be the most relevant. Langlois et al. [2012] proposed to filter features using a forward-backward algorithm to discard linearly correlated features. Using Boosting as learning algorithm, Luong et al. [2015] was able to take advantage of the most significant features.

Finally, several toolkits for WCE were recently proposed: *TranscRater* for ASR [Jalalvand et al., 2016] [1], Marmot for MT[2] as well as WCE toolkit [Servan et al., 2015] [3] that will be presented in more details in the next chapter.

3.2 Granularity of Confidence Estimation (CE)

Confidence Estimation (CE) is the task used to predict the quality of Machine Translation hypotheses given the source sentences. There are various levels in CE depending on the use-cases and the applications such as word-based level CE, phrase-based level CE, sentence-based level CE and document-based level CE that are defined as follows:

- Word-based level Confidence Estimation (WCE): in this task, the aim is to measure the confidence score which is the probability of each word in MT candidates to be a correct translation. In other words, the purpose of this task is to predict the word-level errors in MT hypotheses.
- Phrase-based level Confidence Estimation, also named as Segment-based level CE: the purpose is to measure the quality of distinct phrases in MT output. These phrases could be Noun Phrase, Verb Phrase, Adverbial Phrase, etc.
- Sentence-based level Confidence Estimation: its purpose is to measure the quality of the whole hypothesis sentence of MT output.
- Document-based level Confidence Estimation: its goal is to predict quality of units larger than sentences (entire documents).

[1] `https://github.com/hlt-mt/TranscRater`
[2] `https://github.com/qe-team/marmot`
[3] `https://github.com/besacier/WCE-LIG`

3.3 WCE System for SLT

Given signal x_f in the source language, spoken language translation (SLT) consists in finding the most probable target language sequence $\hat{e} = (e_1, e_2, ..., e_N)$ so that

$$\hat{e} = \underset{e}{\operatorname{argmax}}\{p(e|x_f, f)\} \tag{3.1}$$

where $f = (f_1, f_2, ..., f_M)$ is the transcription of x_f. Now, if we perform confidence estimation at the "words" level, this problem is also named as Word-level Confidence Estimation (WCE) and we can represent this information as a sequence q (same length N of $\hat{e}$) where $q = (q_1, q_2, ..., q_N)$ and $q_i \in \{good, bad\}$[4].

Then, integrating automatic quality assessment in our SLT process can be done as following:

$$\hat{e} = \underset{e}{\operatorname{argmax}} \sum_{q} p(e, q|x_f, f) \tag{3.2}$$

$$\hat{e} = \underset{e}{\operatorname{argmax}} \sum_{q} p(q|x_f, f, e) * p(e|x_f, f) \tag{3.3}$$

$$\hat{e} \approx \underset{e}{\operatorname{argmax}}\{\max_{q}\{p(q|x_f, f, e) * p(e|x_f, f)\}\} \tag{3.4}$$

In the product of 3.4, the SLT component $p(e|x_f, f)$ and the WCE component $p(q|x_f, f, e)$ contribute together to find the best translation output $\hat{e}$. In previous work, WCE has been treated separately in ASR or MT contexts.

3.4 Features Set for WCE in SLT

In this section, we will present a discussion of various state-of-the-art features described in previous works and then inherited in our proposed list of extracted features whose detailed analysis will follow in Section 4.3.

Generally, the features for Word-level Confidence Estimation (WCE) can be classified in two types regarding their origins: the *"internal features"* and the *"external features"*

[4] q_i could be also more than 2 labels, or even scores but this paper mostly deals with error detection (binary set of labels), with the exception of Chapter 6 where three labels are considered.

[Servan et al., 2015]. On the one hand, *internal features* are extracted from the ASR/SMT system itself like language model, alignment table, *N*-best list, word graph, *etc.* On the other hand, *external features* mainly come from linguistic knowledge sources like Part-Of-Speech (POS) Tagger (TreeTagger [Schmid, 1994]), semantic parser (such as DBnary API [Sérasset, 2014] or BabelNet API [Navigli and Ponzetto, 2012]), *etc.*

3.4.1 WCE Features for Speech Transcription (ASR)

Recent works in regarding effective confidence measures have tried to detect errors on ASR outputs. Confidence measures are introduced for Out-Of-Vocabulary (OOV) detection by Asadi et al. [1990].

Young [1994] introduces the use of word posterior probability (WPP) as a confidence measure for speech recognition. It is computed by dividing the total of the posterior probabilities of all hypotheses of the word lattice containing the given word by the total of the posterior probabilities of all word lattice hypotheses in lattice-base search graph [Wessel et al., 2001] [Alabau et al., 2007].

Mauclair et al. [2006] proposed a combination of WPP with Backoff behavior of *N*-gram. The experimental results of this paper showed a significant improvement of CE on the correctness of recognized words.

Also, more recent approaches [Lecouteux et al., 2009] for OOV detection use side-information extracted from the recognizer: hypothesis density, normalized likelihoods (WPP), decoding process behavior, linguistic features, acoustic features (acoustic stability, duration features) and semantic features.

Jalalvand et al. [2016][5] proposed one of the prominent external features for ASR Quality Estimation: Part-Of-Speech (Lexical Features) which indicates grammatical property of each token. Part-Of-Speech (POS) is also named as word class, lexical class, or lexical category. For instance, English POS are verb, pronoun, noun, adjective, adverb, pronoun, preposition, *etc.*

[5] `https://github.com/hlt-mt/TranscRater`

3.4.2 WCE Features for Machine Translation (MT)

3.4.2.1 Internal Features

Xiong et al. [2010] also proposed to use the information of the target token itself and the information of the bigram sequence and the trigram sequence. Moreover, Luong et al. [2013b] proposed the source word features that are the source context aligned to the target token in IOB format (short for Inside, Outside, Beginning). In some cases, one source word could be aligned to many target words. Thus, "B-" prefix and "I-" prefix will be added to the first context of aligned word and the remaining context of aligned words, respectively. "O-" prefix will be added to the context of source word that have no any alignment to any target word.

f	B-je	B-verrai	I-verrai	B-peter	B-demain
e	i	will	see	peter	tomorrow

Table 3.1: Example for IOB format.

Han et al. [2013] focused on various N-gram combinations of target words. Raybaud et al. [2011] described Backoff behaviour of the N-gram (using a target language model) that concentrated on several cases of the occurrences of the previous words depending on the language model.

Moreover, Bach et al. [2011] proposed another internal feature corresponding to the collocations of the target words and source words, also named as the alignment context feature. Table 3.2 presents an example of the alignment context feature in which French, English are the source language and the target language, respectively.

f	les chirurgiens	de	los	angeles	ont	dit
e	surgeons	in	los	angeles	have	said

Table 3.2: Example for the Collocations of target tokens and source tokens.

It can be seen from the data in Table 3.2 that source word *"angeles"* is aligned to target word *"angeles"*. We have thus the values of the source alignment context features (1-word context) such as *"angeles/los", "angeles/angeles", "angeles/ont"* and the values

of the target alignment context features (2-word context) such as *"angeles/in", "angeles/los", "angeles/angeles", "angeles/have", "angeles/said"*.

Ueffing et al. [2003a] presented Word Posterior Probability (WPP), a probability distribution of each target word in the best hypothesis. WPP could be calculated by word graph, *N*-best list.

In addition, Blatz et al. [2004] presented WPP "any" and WPP "exact". WPP "any" of a word in the best hypothesis is conditional probability distribution on all MT candidates containing this word in any position. Its WPP "exact" is calculated by the condition on MT candidates having this word in the same position. Blatz et al. [2004] also showed that the combination of WPP "any" and WPP "exact" has better performance than all the other single features, including heuristic and semantic features.

3.4.2.2 External Features

Blatz et al. [2004], Bach et al. [2011] proposed a lexical feature for MT Confidence Estimation based on Part-Of-Speech (POS). In addition, POS could be tagged by several POS taggers such as TreeTagger [Schmid, 1994], Stanford POS Tagger [Toutanova et al., 2003], Trigrams'n'Tags [Brants, 2000], *etc.*

Bicici [2013] presented the most dominant among several word feature types: "common cover links" (Concerning subtree structure of syntactic tree, the links part from this leaf node of word to other leaf nodes).

Furthermore, Luong et al. [2013b], Bojar et al. [2015] integrated a number of new indicators relying on pseudo reference, syntactic behavior (constituent label, distance to the semantic tree root) and polysemy characteristic. Furthermore, the authors also proposed lexical features whether target token is a stopword, a punctuation mark, a proper noun, a number and semantic feature such as the number of senses of the target and source tokens in WordNet [Miller, 1995].

3.5 Machine Learning Methods

In this section, we will describe a set of effective algorithms to tackle WCE as Naïve Bayes methods, Decision Tree method, Conditional Random Fields (CRFs) technique and Boosting method (concentrating on *AdaBoost* technique).

3.5.1 Naïve Bayes

Naïve Bayes methods are a set of supervised learning techniques based on Bayes theorem with "naïve" independence assumption between each pair of input features. Naïve Bayes methods have several practical applications such as multi-class prediction [Rish, 2001], text classification [McCallum and Nigam, 1998] [Frank and Bouckaert, 2006], spam filtering [Metsis and et al., 2006], sentiment analysis [Pang et al., 2002] [Troussas et al., 2013], real-time prediction [Stella and Amer, 2012], recommendation system [Miyahara and Pazzani, 2000] [Wang and Tan, 2011].

Mathematically, given a possible class outcomes y and a dependent input feature vector $x = \langle x_1, x_2, x_3, ..., x_n \rangle$. Using Bayes' theorem, the conditional probability could be expressed as follows:

$$P(y|x) = P(y|x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n|y)P(y)}{P(x_1, x_2, ..., x_n)} \tag{3.5}$$

Applying the following "naïve" independence assumption for all $x_i \in x$:

$$P(x_i|y, x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y) \tag{3.6}$$

to equation 3.5, we have:

$$P(y|x) = P(y|x_1, x_2, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, x_2, ..., x_n)} \tag{3.7}$$

The goal is to train a classifier $P(y|x)$ that computes the probability distribution over possible value of y given x. In other words, it combines the prior probability with observed data [Mitchell, 1997].

Given a new instance $x = \langle x_1, x_2, x_3, ..., x_n \rangle$, we can find the most probable candidate of y using the naïve Bayes classification rule:

$$\hat{y} = \underset{y}{\operatorname{argmax}} \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(x_1, x_2, ..., x_n)} = \underset{y}{\operatorname{argmax}}\, P(y) \prod_{i=1}^{n} P(x_i|y) \tag{3.8}$$

Here $P(x_1, x_2, ..., x_n)$ is constant given the input, thus could be ignored; $P(y)$ denotes the relative frequency of y in given training corpora; the likelihood of the features $P(x_i|y)$ could be generated by different models such as a Normal distribution, also called Gaussian naïve Bayes:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{3.9}$$

where σ_y and μ_y are standard deviation and mean varying from feature to feature, respectively.

3.5.2 Decision Tree

Decision Tree method [Quinlan, 1986] is one of the methods commonly used in data mining, statistics, machine learning and natural language processing domains. It is represented by a tree structure in which each internal node corresponds to an attribute, each branch from a node denotes a value of an attribute, the topmost node represents the root node of the tree. For instance, Figure 3.1 shows a decision tree representation.

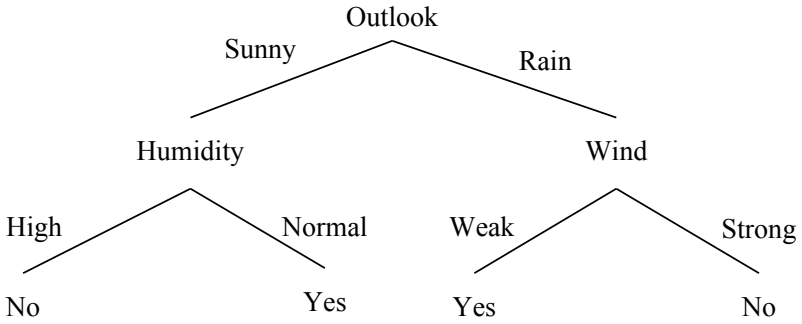

Figure 3.1: An Example of Decision Tree Technique.

In addition, given training data is represented in records of the following form:

$$(x, Y) = (x_1, x_2, ..., x_n, Y) \tag{3.10}$$

where $x = (x_1, x_2, ..., x_n)$ is an instance of the set of possible instances X.

For example, in Figure 3.1, we have x = ⟨Outlook="Sunny", Humidity="High"⟩;

(x, Y) = ⟨Outlook="Sunny", Humidity="High", PlayTennis="No"⟩.

Moreover, we could have the set of function that illustrate the hypotheses:

$$T = \{t | t : X \rightarrow Y\} \tag{3.11}$$

where Y is a set of discrete values or a set of continuous values; each hypothesis t is denoted by a decision tree.

Note that, if Y takes discrete values (a finite set of values), we call decision tree models as "classification tree" using the metrics *entropy* and *information gain* to find the "best" decision attribute such as ID3 (Iterative Dichotomiser 3) [Quinlan, 1986], C4.5 [Quinlan, 1993], C5.0 (an evolved version of C4.5). And if Y takes continuous values, they are called by "regression tree" using the metrics *Gini impurity* to find the "best" decision attribute such as CART (Classification And Regression Trees) [Breiman et al., 1984].

In practice, in order to construct more than one decision tree, we could use the following strategies:

- Random decision forests classifier [Ho, 1995, 1998]
- Boosted tree technique that emphasizes the training instances of previous 'weak learners', such as Adaptive Boosting (Adaboost) that will be detailed in Subsection 3.5.4
- Bagging decision tree [Breiman, 1996].

Moreover, to reduce the size of decision trees, we could use "pruning" technique [Mansour, 1997]. In other words, this technique is used to remove nodes of the decision tree

that provides little supplementary information. There are several techniques to prune the sub-trees beginning from the root of the decision tree or starting at its leaf such as Reduced Error Pruning [Quinlan, 1987], Cost Complexity Pruning [Breiman et al., 1984], Error-based Pruning [Quinlan, 1993].

3.5.3 Conditional Random Fields (CRFs)

Conditional Random Fields (CRFs) [Lafferty et al., 2001] are the discriminative probabilistic undirected graphical models used to measure the conditional probability of a label sequence given the observation sequence.

Mathematically, let $X = x_1, x_2, x_3, ..., x_n$ and $Y = y_1, y_2, y_3, ..., y_n$ denote the observation sequence and the label sequence, respectively.

Let $G = (V, E)$ be a probabilistic graph.

Let $V = X \cup Y$ be the set of the probability distributions of the nodes (vertices) that denote the set of the cliques C in the graph G.

Let $E \subseteq V \times V$ denote the set of the edges of the graph G.

Therefore, a CRF defines the conditional probability of the random variable $y \in Y$ conditioned on X as follows,

$$P(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \exp\left(\sum_k \lambda_k f_k(y_c, x_c)\right) \tag{3.12}$$

Here,

- f_k are the feature functions or sufficient statistics on any subset of random variable in the pair $(y_c, x_c) \in (Y, X)$
- λ_k are the real-valued parameter vectors, also known as the trained weights for each feature function. This parameter estimation is typically calculated by maximizing the likelihood function of training data using by gradient descent algorithms or quasi-Newton techniques such as BFGS [Bertsekas, 1999]

- $Z(x)$ denotes the observation-dependent normalization term over all possible state sequence:

$$Z(x) = \sum_{y} \prod_{c \in C} \exp\left(\sum_{k} \lambda_k f_k (y_c, x_c) \right) \tag{3.13}$$

More details of CRF-based models and the relationship between CRF-based models (such as linear-chain CRF, general CRF), naïve Bayes, logistic regression and hidden Markov models are described in [Sutton and McCallum, 2012].

3.5.4 Boosting Method

The main purpose of boosting method [Kearns and Valiant, 1989] [Schapire, 1990] [Freund, 1995] is to build a robust learning technique that is an ensemble of given 'weak' learning algorithms for improving the prediction accuracy.

Algorithm 1 AdaBoost algorithm combining K 'weak' learners rules. D denotes the set of all training data and $H_k, 1 \leq k \leq K$ is the learner function at each step of the algorithm [Freund and Schapire, 1999].

Initialize the weighted for training corpus: $w_i = \frac{1}{N}, i = 1,2,3,...,N$
while $k \leq K$ **do**
Finding 'weak' classifier H_k on given training data D_k after applying current weights w_i.
Calculating the error rate of current classifier: $\varepsilon_k = P_{d_i \sim D_k} [H_k(x_i) \neq y_i]$
Updating the weighted training corpus with the weighted contribution of current classifier $\alpha_k = \frac{1}{2} \ln\left(\frac{1-\varepsilon_k}{\varepsilon_k}\right)$ and Z_k denotes the normalization constant

$$D_{k+1}(i) = \begin{cases} \frac{1}{Z_k} D_k(i) \exp(-\alpha_k) & \text{if } y_i = H_k(x_i), \\ \frac{1}{Z_k} D_k(i) \exp(\alpha_k) & \text{otherwise} \end{cases}$$

end while
Generating the final classifier that combines above 'weak' classifiers:

$$H(x) = \text{sign}\left(\sum_{k=1}^{K} \alpha_k H_k(x) \right)$$

In our work, we use AdaBoost (Adaptive Boosting) [Freund and Schapire, 1996] that is one of the most well-known boosting methods. The differences between AdaBoost and

other boosting methods are that the 'weak' classifiers are learned on weighted training data whose weights are generated from previous classifier or the initialized weights.

AdaBoost is described by Algorithm 1 with given training data $D = \{d_1, d_2, ..., d_N\} = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, K is the maximum number of classifiers in ensemble method.

In summary, when finding the learner rules, whereas boosting method uses random subset training data, AdaBoost utilizes the weighted training data [Ferreira and Figueiredo, 2012].

3.6 Evaluation

This section presents automatic performance metrics for classification (binary label or multi-label) in pattern recognition and information retrieval [Rijsbergen, 1979] that are *recall* (also called as sensitivity), *precision* (so-called as positive predictive value) and *F-measure* (harmonic mean of *recall* and *precision*) used for this thesis. We also describe briefly other metrics such as Mean Absolute Error (MAE)and Root Mean Square Error (RMSE).

3.6.1 Precision, Recall, F-measure

In this study, *recall* and *precision* of label "G" are given as follows,

$$precision = \frac{|\{relevant\ tokens\} \cap \{retrieved\ tokens\}|}{\{retrieved\ tokens\}} \times 100\% \quad (3.14)$$

$$recall = \frac{|\{relevant\ tokens\} \cap \{retrieved\ tokens\}|}{\{relevant\ tokens\}} \times 100\% \quad (3.15)$$

While $\{retrieved\ tokens\}$ is the number of tokens predicted as "G", $\{relevant\ tokens\}$ denotes the number of tokens whose oracle labels are "G". It is noted that the numerators in two Equations 3.14 and 3.15 denote the number of tokens that both its oracle label and real label assigned by the classification system are "G".

Therefore, *precision* illustrates how many returned labels are correct while *recall* shows how many relevant labels the model could return.

Generally, used as a mean of ratios, *harmonic mean* is defined as follows,

$$H_{mean} = \frac{N}{\sum_{i=1}^{N} \frac{1}{m_i}} \times 100\% \tag{3.16}$$

When $N = 2$, $m_1 = precision$ and $m_2 = recall$, thus, we have:

$$\begin{aligned} H_{mean} &= \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \times 100\% \\ &= \frac{2 \times precision \times recall}{precision + recall} \times 100\% \end{aligned} \tag{3.17}$$

H_{mean} also known as *F-measure* [Rijsbergen, 1979] should satisfy the constraint,

$$0 \leq F\text{-}measure \leq 100\% \tag{3.18}$$

So, to assess *F-measure*, we could use the next formula,

$$F\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \times 100\% \tag{3.19}$$

Similarly, to estimate the performance metrics (*precision*, *recall* and *F-measure*) of other labels (for example, label "B"), we could reuse the Equations 3.14, 3.15 and 3.19.

3.6.2 Mean Absolute Error (MAE), Root Mean Square Error (RMSE)

To estimate the quality of the performance of sentence-level CE system, we could use two of the most common metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE).

The mean absolute error (MAE) is a metric used to assess how far prediction scores differ from oracle scores.

The root mean square error (RMSE), also called the root mean square deviation (RMSD), measures the average difference between prediction scores and oracle scores.

Mathematically, let $y = (y_1, y_2, ..., y_N)$ and $\hat{y} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_N)$ denote the prediction scores and the oracle scores of the test corpora having N sentences, respectively.

$$MAE = N^{-1} \sum_{t=1}^{N} |e_t| = N^{-1} \sum_{t=1}^{N} |y_t - \hat{y}_t| \tag{3.20}$$

$$RMSE = \sqrt{N^{-1} \sum_{t=1}^{N} e_t^2} = \sqrt{N^{-1} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2} \tag{3.21}$$

Furthermore, both MAE and RMSE estimate the mean of forecast error in distributions of variable in test sample using negatively oriented scores. It means that lower values are better. [Willmott and Matsuura, 2005] presented the clear comparisons between MAE and RMSE in assessing average model performance.

3.7 Conclusion

In this chapter, we described an overview of quality estimation in spoken language translation. We described several CE approaches. We began exploring several conventional features (also named as *'prediction indicators'*). They are proposed for the quality estimation of both Automatic Speech Recognition system and Machine Translation system and inheriting thus in our proposed list of features for SLT in the following chapter. Next, we reviewed a few ML techniques used to solve WCE problems for SLT and to optimize models with the aim of improving the prediction performance. Performance metrics for WCE were also introduced.

In the next chapter, we present the main methods used in our investigation. Furthermore, we will also propose our *LIG-WCE Toolkit* which is a complete out-of-the-box WCE system for SLT and we will show the preliminary results as well.

Chapter 4

An Evaluation Framework for Confidence Estimation in Spoken Language Translation

4.1 Motivation

In this chapter, we focus on presenting the experimental setup and the main components of WCE system for SLT to build preliminary results as well. We propose both the formalization of WCE system for SLT and a complete out-of-the-box WCE system, as well as home made corpus. This represents a complete evaluation framework for reproducible experiments in SLT confidence estimation.

The remainder of this chapter is structured as follows. Section 4.2 describes speech corpus (distributed to the research community[1]) dedicated to WCE for SLT and presents the experimental setup as well. In Section 4.3, we propose our *LIG-WCE toolkit* to predict the quality of Spoken Language Translation output that integrates several existing libraries / toolkits to extract the list of novel features for SLT system inherited the conventional ones presented in Chapter 3. We then detail and analyze the preliminary

[1] `https://github.com/besacier/WCE-SLT-LIG`

results using only MT features or only ASR features in Section 4.4. Finally, Section 4.5 concludes this chapter[2].

4.2 Dataset, ASR and MT Modules

4.2.1 Dataset

4.2.1.1 Starting Point: MT Post-Edited Corpus

We applied our SMT system for French-English translation task. It generates from 10881 French sentences (News corpora of evaluation campaign from 2006 to 2010 in Workshop on Machine Translation) to English hypotheses. The baseline SMT system will be presented in subsection 4.2.3. Post-edition corpus was collected by a crowdsourcing platform (Amazon's Mechanical Turk) [Potet et al., 2012]. Note that Potet et al. [2012] showed that more than 87% of collected post-editions was judged to improve the hypotheses and more than 94% of the crowdsourced post-editions was assessed at least of professional quality.

To label each target word, we used TERp-A toolkit [Snover et al., 2008]. Table 4.1 presents the labels obtained using TERp-A toolkit for one reference (post-edition) and hypothesis pair. Each phrase or word in hypothesis is aligned to a phrase of word of the reference with various types such as substitution("S"), phrasal substitution ("P"), insertion ("I"), stem matches ("T") and synonym matches ("Y"). Moreover, to mark an exact match, we used symbol "E". Therefore, to apply binary classifiers (*good*/*bad*), we separate above symbols into 2-label set: Y, T, E belong to *good* label set while I, P, S belong to *bad* label set.

[2]Many of the findings observed in this chapter were published in [Le et al., 2016a] and in [Servan et al., 2015]

Reference	The	consequence	of	the	fundamentalist	
	E	S	E	E	S	
Hyp After Shift	The	result	of	the	hard-line	
Reference	movement		also	has	its importance	.
	Y	I	E	D	P	E
Hyp After Shift	trend	is	also		important	.

Table 4.1: Example of labels extracted by TERp-A toolkit.

4.2.1.2 Extending the Corpus with Speech Recordings and Transcripts

The *dev* set and *tst* set of this corpus were recorded by french native speakers. Each sentence was uttered by 3 speakers, leading to 2643 and 4050 speech recordings for *dev* set and *tst* set, respectively. For each speech utterance, a quintuplet containing: ASR output (f_{hyp}), verbatim transcript (f_{ref}), English text translation output ($e_{hyp_{mt}}$), speech translation output ($e_{hyp_{slt}}$) and post-edition of translation (e_{ref}), was made available. This corpus is available on a *github* repository[3]. More details are given in Table 4.2. The total length of the *dev* and *tst* speech corpora obtained are 16h52, since some utterances were pretty long. Next sections detail how this quintuplet was obtained using ASR and MT.

Corpus	#sentences	#speech recordings	#speakers	Duration
dev	881	2643	15 (9 women + 6 men)	5h51
tst	1350	4050	27 (11 women + 16 men)	11h01

Table 4.2: Details on our *dev* and *tst* corpora for SLT.

4.2.2 ASR Systems

To obtain the speech transcripts (f_{hyp}), we built a French ASR system based on KALDI toolkit [Povey et al., 2011b]. Acoustic models are trained using several corpora (ESTER, REPERE, ETAPE and BREF120) representing more than 600 hours of french transcribed speech.

[3] `https://github.com/besacier/WCE-SLT-LIG/`

LM	**1-gram**	**2-grams**	**3-grams**
small (*ASR*1)	62K	1M	59M
big (*ASR*2)	95K	49M	301M

Table 4.3: Details on language models (LM) used in our two ASR systems.

The baseline GMM-HMM system is based on mel-frequency cepstral coefficient (MFCC) acoustic features (13 coefficients expanded with delta and double delta features and energy: 40 features) with various feature transformations including linear discriminant analysis (LDA), maximum likelihood linear transformation (MLLT), and feature space maximum likelihood linear regression (fMLLR) with speaker adaptive training (SAT). The GMM acoustic model makes initial phoneme alignments of the training data set for the following DNN acoustic model training.

The speech transcription process is carried out in two passes: an automatic transcript is generated with a GMM-HMM model of 43182 states and 250000 Gaussians. Then word graphs outputs obtained during the first pass are used to compute a fMLLR-SAT transform on each speaker. The second pass is performed using DNN acoustic model trained on acoustic features normalized with the fMLLR matrix. CD-DNN-HMM acoustic models are trained (43182 context-dependent states) using GMM-HMM topology.

We propose to use two 3-gram language models trained on French ESTER corpus [Galliano et al., 2006] as well as on French Gigaword (vocabulary size are respectively 62k and 95k). The ASR systems LM weight parameters are tuned through WER on the *dev* corpus. Details on these two language models can be found in Table 4.3.

In our experiments, we propose two ASR systems based on the previously described language models. The first system (*ASR*1) uses the small language model allowing a fast ASR system (about 2x Real Time), while in the second system lattices are rescored with a big language model (about 10x Real Time) during a third pass.

Table 4.4 presents the performances obtained by two above ASR systems.

These WER may appear as rather high according to the task (transcribing read news).

Task	***dev* set**	***tst* set**
ASR1	21.86%	17.37%
ASR2	16.90%	12.50%

Table 4.4: ASR performance (WER) on our *dev* and *tst* set for the two different ASR systems.

A deeper analysis shows that these news contain several foreign-named entities, especially in our *dev* set. This part of the data is extracted from French medias dealing with european economy in EU. This could also explain why the scores are significantly different between *dev* and *tst* sets. In addition, automatic post-processing is applied to ASR output in order to match requirements of standard input for MT system.

4.2.3 SMT System

We use *Moses* toolkit, phrase-based translation system, [Koehn et al., 2007] to translate French ASR into English (e_{hyp}). We also use some scripts, given by *Moses* toolkit, to lowercase, to normalize, to tokenize, to calculate BLEU score such as *lowercase.perl, normalize-punctuation.perl, tokenizer.perl, multi-bleu.perl*, respectively.

To train our target language model, we use SRI Language Modeling (SRILM) Toolkit [Stolcke, 2002] on News monolingual corpus (48653884 sentences). We use News and Europarl parallel corpus (1638440 sentences) using for WMT evaluation campaign 2010 to train our target translation model. In addition, we also keep the values of default configuration when running *Moses* toolkit: log-linear model with 15 weighted component scores, including that 6 lexical reordering, 1 distortion, 1 language model, 1 word penalty, 1 phrase penalty, 4 translation model, 1 unknown word penalty [Potet et al., 2010].

In the decoder phase, we used the following options to generate the information of both source and target languages such as N-best hypotheses of SMT system, word alignment information in N-best list:

- *-include-segmentation-in-n-best* and *-print-alignment-info-in-n-best*: extract the information of word-to-word alignments in the N-best list; noted that word-to-word alignments are excluded from the phrase table.

- *-n-best-list PATH_OF_FILE SIZE [distinct]*: extract an N-best hypotheses of size *SIZE* to the path of file *PATH_OF_FILE*.

4.2.4 Obtaining Quality Assessment Labels for SLT

After building an ASR system, we have a novel factor of quintuple: ASR hypothesis f_{hyp}. Its reference version is our verbatim transcript called f_{ref}. After translating ASR output (f_{hyp}) by the same SMT system (already mentioned in subsection 4.2.3), we have new translation output, called $e_{hyp_{slt}}$. Note that $e_{hyp_{slt}}$ is a degraded version of translation of f_{ref} ($e_{hyp_{mt}}$).

To obtain word label setting for WCE, we used TERp-A toolkit [Snover et al., 2008] between speech translation output ($e_{hyp_{slt}}$) and post-editions obtained from the text translation task (e_{ref}). Therefore, we re-used initial post-edition to infer labels of a SLT task. Table 4.7 and Table 4.8 present MT and SLT performances on our corpus.

The above-mentioned remark makes the value of this corpus. For example, we can obtain a quintuplet (ASR hypothesis, verbatim transcript, MT hypothesis, target translation and SLT hypothesis) from TED corpus. However, there are two differences: firstly, to deal with speaker variability and different ASR hypotheses for a specific sentence, each sentence is recorded by three different speakers; secondly, the target translation of TED is not post-editing version of an automatic translation because it is a manual translation of prior subtitles and it is not possible to guarantee that *good/bad* labels generated from this would be reliable for WCE [Besacier et al., 2014].

4.2.5 Summary Statistics of Corpus

Table 4.5 presented the summary statistics of our corpus. In which, we show how to obtain WCE labels. To evaluate WCE for 3 tasks, we have all data:

- **ASR**: generate *good*/*bad* labels by calculating WER between f_{hyp} and f_{ref},
- **MT**: generate *good*/*bad* labels by calculating TERp-A between $e_{hyp_{mt}}$ and e_{ref},
- **SLT**: generate *good*/*bad* labels by calculating TERp-A between $e_{hyp_{slt}}$ and e_{ref}.

Data	**# *dev* utt**	**# *tst* utt**	**# *dev* words**	**# *tst* words**	**method to obtain WCE labels**
f_{ref}	881	1350	21 988	36 404	
f_{hyp1}	881*3	1350*3	66 435	108 332	wer(f_{hyp1}, f_{ref})
f_{hyp2}	881*3	1350*3	66 834	108 598	wer(f_{hyp2}, f_{ref})
$e_{hyp_{mt}}$	881	1350	22 340	35 213	terpa($e_{hyp_{mt}}$, e_{ref})
$e_{hyp_{slt1}}$	881*3	1350*3	61 787	97 977	terpa($e_{hyp_{slt1}}$, e_{ref})
$e_{hyp_{slt2}}$	881*3	1350*3	62 213	97 804	terpa($e_{hyp_{slt2}}$, e_{ref})
e_{ref}	881	1350	22 342	34 880	

Table 4.5: Overview of Post-edited Corpus for SLT.

Table 4.6 gives an example of quintuplet in our corpus. While f_{hyp1} (transcript) has 1 error, f_{hyp2} has 4. Therefore, this points out 2 *bad* labels ($e_{hyp_{slt1}}$) and 4 *bad* labels ($e_{hyp_{slt2}}$) in speech translation hypothesis while $e_{hyp_{mt}}$ has only 1 *bad* label.

f_{ref}	quand	notre	cerveau	chauffe
f_{hyp1}	*comme*	notre	cerveau	chauffe
labels ASR	B	G	G	G
f_{hyp2}	*qu'*	*entre*	*serbes*	*au* chauffe
labels ASR	B	B	B	B G
$e_{hyp_{mt}}$	when	our	brains	*chauffe*
labels MT	G	G	G	B
$e_{hyp_{slt1}}$	*as*	our	brains	*chauffe*
labels SLT	B	G	G	B
$e_{hyp_{slt2}}$	*between*	*serbs*	*in*	*chauffe*
labels SLT	B	B	B	B
e_{ref}	when	our	brain	heats up

Table 4.6: Example of quintuplet with associated labels.

Table 4.7 and Table 4.8 summarize baseline ASR, MT and SLT performances on our corpus as well as the distribution of the binary labels (*good*, *bad*) extracted for both tasks. Normally, in same condition, percentage of *bad* labels is decreased from SLT to MT task.

Task	**ASR (WER)**	**MT (BLEU)**	**% G (*good*)**	**% B (*bad*)**
MT	0%	49.13%	76.93%	23.07%
SLT (ASR1)	21.86%	26.73%	62.03%	37.97%
SLT (ASR2)	16.90%	28.89%	63.87%	36.13%

Table 4.7: MT and SLT performances on our *dev* set.

Task	**ASR (WER)**	**MT (BLEU)**	**% G (*good*)**	**% B (*bad*)**
MT	0%	57.87%	81.58%	18.42%
SLT (*ASR*1)	17.37%	36.21%	70.59%	29.41%
SLT (*ASR*2)	12.50%	38.97%	72.61%	27.39%

Table 4.8: MT and SLT performances on our *tst* set.

4.3 LIG-WCE Toolkit

4.3.1 Motivation

Recently, a growing need of Confidence Estimation (CE) for both Statistical Machine Translation (SMT) systems and Automatic Speech Recognition (ASR) system in Computer Aided Translation (CAT), interactive speech to speech translation, was observed. However, most of CE toolkits are optimized for a single target language (mainly English) and, as far as we know, none of them are dedicated to this specific task and freely available.

Our experience in participating in *task 2* (WCE - shared task of the WMT (Workshop on Machine Translation)) leads us to the following observation: while feature processing is very important to achieve good performance, it requires to call a set of heterogeneous Natural Language Processing tools (for lexical, syntactic, semantic analyses).

Therefore, the main purpose of *LIG-WCE Toolkit* is to unify the feature processing, together with the call of machine learning algorithms, to facilitate the design of confidence estimation systems. In other words, we propose a method that could point out both correct and incorrect parts in SLT output. In addition, we propose *LIG-WCE Toolkit*, as an open-source toolkit for forecasting the words' quality of SLT hypothesis, whose novel

contributions are (i) support for various target languages, (ii) handle a number of features of different types (system-based, lexical, syntactic and semantic) of both SMT system and ASR system. Our toolkit also integrates a wide variety of NLP or ML tools to pre-process data, extract features and estimate confidence at word-level. Features for Word-level Confidence Estimation (WCE) can be easily added / removed using a configuration file.

4.3.2 Formalization

We propose to build an efficient quality assessment (WCE) system with the goal of assessing the quality estimation (or error detection) component in speech translation by the following equation:

$$\hat{q} = \underset{q}{\operatorname{argmax}}\{p_{SLT}(q|x_f, f, \hat{e})\} \tag{4.1}$$

where x_f is the given signal in the source language; $\hat{e}$[4] $= (e_1, e_2, ..., e_N)$ is the most probable target language sequence from the spoken language translation (SLT) process ; $f = (f_1, f_2, ..., f_M)$ is the transcription of x_f; $q = (q_1, q_2, ..., q_N)$ is a sequence of error labels on the target language and $q_i \in \{good, bad\}$[5]. This is a sequence labeling task that can be solved with several machine learning techniques such as Conditional Random Fields (CRF) [Lafferty et al., 2001]. However, for that, we need a large amount of training data for which a quadruplet (x_f, f, e, q) is available.

As it is much easier to obtain data containing either the triplet (x_f, f, q) (ASR output + manual references and error labels inferred from WER) or the triplet (f, e, q) (MT output + manual post-editions and error labels inferred using tools such as TERp-A [Snover et al., 2008]) we can also recast error detection with the following equation:

$$\hat{q} = \underset{q}{\operatorname{argmax}}\{p_{ASR}(q|x_f, f)^{\alpha} * p_{MT}(q|e, f)^{1-\alpha}\} \tag{4.2}$$

where α is a weight giving more or less importance to error detector on transcription WCE_{ASR} (quality assessment on transcription) compared to error detector on translation

[4] written simply e for convenience in any other equations.

[5] at this point q_i takes two values (G/B) but will evolve to 3 labels later on in Chapter 6.

WCE_{MT} (quality assessment on translation). It is important to note that $p_{ASR}(q|x_f, f)$ corresponds to the quality estimation of the words in the target language based on features calculated on the source language (ASR). For that, what we do is projecting source quality scores to the target using word alignment information between e and f sequences. This alternative approach (Equation 4.2) will be also evaluated in this work even if it corresponds to a different optimization problem than Equation 4.1.

In both approaches – *joint* ($p_{SLT}(q|x_f, f, e)$) and *combined* ($p_{ASR}(q|x_f, f)$ + $p_{MT}(q|e, f)$) – some features need to be extracted from ASR and MT modules. They are more precisely detailed in next subsections.

4.3.3 WCE Features for Speech Transcription (ASR)

In this task, we generate various categories of features which are extracted from scores of language model, from syntactic or morphological analysis, from ASR graph. They are described below:

- Acoustic features: word duration (**F-dur**).
- Graph features (extracted from the ASR word confusion networks): number of alternative (**F-alt**) paths between two nodes; word posterior probability (**F-post**).
- Linguistic features (based on probabilities by the language model): word itself (**F-word**), 3-gram probability (**F-3g**), log probability (**F-log**), back-off level of the word (**F-back**), as proposed in [Fayolle et al., 2010],
- Lexical Features: Part-Of-Speech (POS) of the word (**F-POS**),
- Context Features: Part-Of-Speech tags in the neighborhood of a given word (**F-context**). Note that **F-context** features are formed by its content (**F-word**) and one POS before (left **F-POS**) or one POS after (right **F-POS**) the source word. With the example presented in Table 4.9, **F-POS** of the source word “indépendance” (**F-word**) is “NOUN”. Therefore, its **F-context** features are “indépendance/DET:ART”, “indépendance/NOUN” and “indépendance/VERB”.

F-word	la	nature	de	l'	indépendance	octroyée	...
F-POS	DET:ART	NOUN	PRP	DET:ART	NOUN	VERB	...

Table 4.9: Example of **F-context** where source words are aligned to POS.

For each word in the ASR hypothesis, we estimate the 9 features (F-Word; F-3g; F-back; F-log; F-alt; F-post; F-dur; F-POS; F-context) previously described.

In a preliminary experiment, we will evaluate these features for quality assessment in ASR only (WCE_{ASR} task). Two different classifiers will be used: a variant of boosting classification algorithm called *bonzaiboost* [Laurent et al., 2014a] (implementing the boosting algorithm *Adaboost.MH* over deeper trees) and the Conditional Random Fields [Lafferty et al., 2001].

4.3.4 WCE Features for Machine Translation (MT)

Several knowledge sources are employed for generating features, in a total of 24 features, see Table 4.10.

These features were chosen because of their relevance in previous Word-level Confidence Estimation tasks [Callison-Burch et al., 2012] [Bojar et al., 2013] [Bojar et al., 2014]. Some of them are already described in detail in some previous papers [Wessel et al., 2001] [Ueffing et al., 2003b] [Blatz et al., 2004] [Xiong et al., 2010] [Langlois et al., 2012] [Luong et al., 2015] [Raybaud et al., 2011]. Consequently, the novel features, which we added into our current toolkit, are in **"bold"** in Table 4.10. Also, the features in *"italic"* are conventional features but extracted using a new approach.

The feature list could be extended (by us or by other contributors) in the future, since the toolkit is made available to the research community. For instance, we plan to integrate the use of monolingual or bilingual word embeddings following the works of Mikolov et al. [2013b].

It is important to note that we extract features regarding *tokens* in the translated hypothesis (MT or SLT). In other words, one feature is extracted for each token in the MT output. So, in the Table 4.10, *target* refers to the feature coming from the translated

1 *Proper Name*	10 Stop Word	19 WPP max
2 **Unknown Stem**	11 *Word context Alignments*	20 Nodes
3 Num. of Word Occ.	12 *POS context Alignments*	21 Constituent Label
4 **Num. of Stem Occ.**	13 **Stem context Alignments**	22 Distance To Root
5 *Polysemy Count – Target*	14 Longest Target *N*-gram Length	23 Numeric
6 Backoff Behaviour – Target	15 Longest Source *N*-gram Length	24 Punctuation
7 *Alignment Features*	16 WPP Exact	
8 Occur in Google Translate	17 WPP Any	
9 **Occur in Bing Translator**	18 WPP min	

Table 4.10: Features extracted by the toolkit: highlights in **bold** are the new features we propose, the other features are those classically extracted - we put in *italic* those for which we proposed a new extraction method compared to our previous work.

hypothesis and *source* refers to a feature extracted from the source word aligned to the considered target word. More details on some of these features are given in the next subsections.

4.3.4.1 Internal Features

These features are given by the Machine Translation system, which outputs additional data like *N*-best list, word graph.

- **Alignment context features**: these features (#11-13 in Table 4.10) are based on collocations and proposed by Bach et al. [2011]. Collocations could be an indicator to estimate when a target word is aligned by a specific source word. We also apply the reverse, the collocations regarding the source side (#7 in Table 4.10 - simply called **Alignment Features**):
  - ◇ *Features of target alignment context*: the combinations of one source word, one target word (with which it is aligned), and one target word before and one target word after.
  - ◇ *Features of source alignment context*: the combinations of one target word, the source word (with which it is aligned), and one source word before and one source word after (left and right contexts, respectively).

With the example presented in Table 4.11, the target word "of" is aligned with "de". The source context extracted corresponds to the two words around "de", which are "nature" and "l' ". The *source alignment context features* are "of/nature", "of/de" and "of/l' " In the same way, he *target alignment context features* of "de" are: "de/nature", "de/of" and "de/the".

We applied the same context extraction for Part-of-Speech and Stems.

Target	the	nature	of	the	independence	granted	...
Source	la	nature	de	l'	indépendance	octroyée	...

Table 4.11: Example of parallel sentence where words are aligned one-to-one.

- **Longest Target (or Source) N-gram Length**: we seek to get the length $(n+1)$ of the longest left sequence (w_{i-n}) concerned by the current word (w_i) and known by the language model (LM) concerned (source and target sides). For example, if the sequence of words $w_{i-2}w_{i-1}w_i$ occurs in the target LM, the longest target N-gram value for w_i will be 3. This value ranges from 0 to the max order of the LM concerned. We also extract a redundant feature called **Backoff Behavior Target** [Raybaud et al., 2011]. In fact, we extract the backoff behavior features of LM from the backward sequences of each target word. Our toolkit extracts how often, for each word in the target sentence, the LM has to back off to assign a probability to the sentence.

- **Word Posterior Probability** (WPP) and **Nodes** features are extracted from a confusion network, which comes from the output of the Machine Translation N-best list. **WPP Exact** is the WPP value for each word concerned at the exact same position in the graph. **WPP Any** extracts the same information at any position in the graph. **WPP Min** gives the smallest WPP value concerned by the transition and **WPP Max** its maximum.

  In the example shown in Figure 4.1, the target word *"function"* gets a **WPP Exact** at *0.2*, **WPP Min** at *0.1* and **WPP max** at *0.4*.

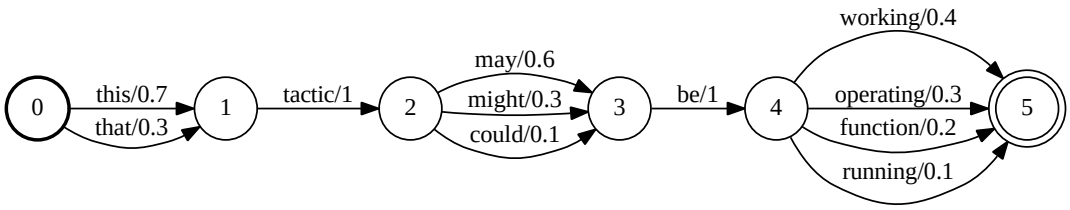

Figure 4.1: Example of Confusion Network

4.3.4.2 External Features

Below is the list of the external features used:

- **Proper Name**: indicates if a word is a proper name (same binary features are extracted to know if a token is **Numerical**, **Punctuation** or **Stop Word**).
- **Unknown Stem**: informs whether the stem of the considered word is known or not.
- **Number of Word/Stem Occurrences**: counts the occurrences of a word/stem in the sentence.
- The target word's constituent label (**Constituent Label**) and its depth in the constituent tree (**Distance to Root**) are extracted using a syntactic parser, Figure 4.2 illustrates the distance between a word and its root in the tree. In the case of *"working"*, the **Constituent Label** is *VBG* and the **Distance to Root** value is *6*.

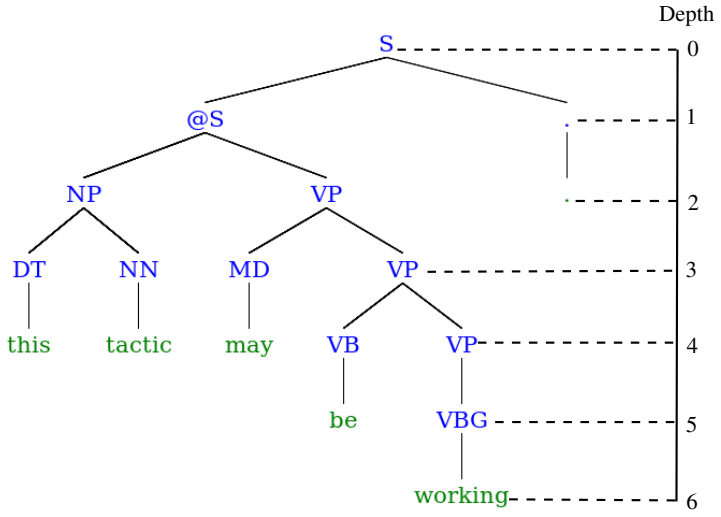

Figure 4.2: Example of constituent tree.

- **Target Polysemy Count**: we extract the polysemy count, which is the number of meanings of a word in a given language.

- **Occurences in Google Translate** and **Occurences in Bing Translator**: in the translation hypothesis, we (optionally) test the presence of the target word in online translations given respectively by *Google Translate* and *Bing Translator*[6].

In this thesis, we will use Conditional Random Fields (CRFs) [Lafferty et al., 2001] as our machine learning technique. Also, we apply WAPITI toolkit [Lavergne et al., 2010] to train our WCE estimator based on both MT and ASR features.

4.3.5 Our Proposed Toolkit

In this section, we detail our toolkit, which is a complete out-of-the-box Word-level Confidence Estimation (WCE) system. It is a customizable, flexible, and portable platform.

4.3.5.1 Pipeline Overview

Our toolkit is described in Figure 4.3. It contains three essential components: *preprocessing*, *feature extraction* and *training / labeling*. It integrates several existing Natural Language Processing (NLP) tools and API. It is developed in *Python 3* to use efficiently existing libraries/toolkits as well as being object-oriented designed.

The source code is available on a *GitHub* repository[7] and provided with ready-made scripts to run reproducible experiments on a French–English WCE task (for which the data is also made available).

[6]Using this kind of feature is controversial, however we observed that such features are available in general use case scenarios, so we decided to include them in our experiments. Contrastive results without these 2 features will be also given later on.

[7]`https://github.com/besacier/WCE-LIG`

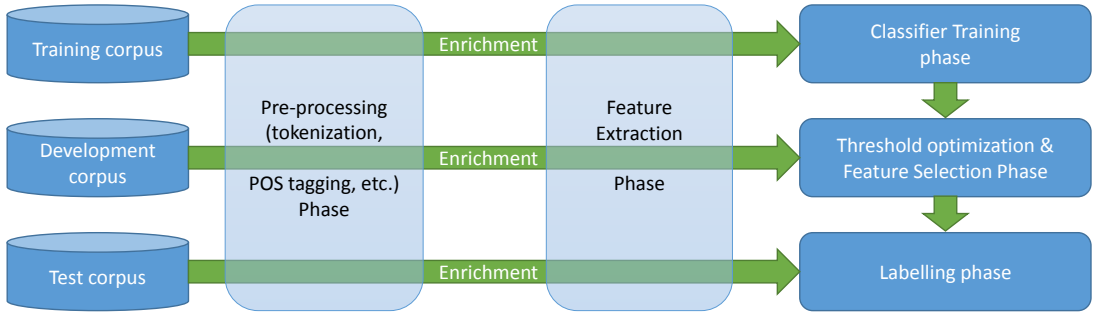

Figure 4.3: Pipeline of our Word-level Confidence Estimation tool.

4.3.5.2 System Design

The first steps are the preprocessing and the feature extraction during which the toolkit processes and adds information to the initial corpora available. Then, the most important step consists of training a classifier using the features extracted (training phase) or in the labelling of the test corpus (decoding phase).

We also added a threshold optimization and a feature selection phase which are later described (see Sections 5.1 and 5.2 respectively for threshold optimization and feature selection).

All these phases can be parameterized using a single configuration file.

4.3.5.3 System Configuration

A configuration file gathers the main WCE parameters. It is stored in YAML[8] format. The main configuration parameters concern the source and target languages involved and the path to the input corpus and its translation.

4.3.5.4 Preprocessing Phase

Preprocessing consists of obtaining POS tags, word alignments and all needed analyses from the available parallel corpus (the target being a MT output made up of raw text – 1-best and *N*-best of MT). First, input data is lowercased and/or tokenized if necessary.

[8] `http://www.yaml.org/`

Then, TreeTagger toolkit [Schmid, 1995] is applied to get the Part-Of-Speech (POS) tags and stem of each word in both source and target languages. The different POS extracted are normalized. Finally, word alignments are obtained using GIZA++ [Och and Ney, 2003].

4.3.5.5 Features Extraction

As said before, the internal features come from the output of the Statistical Machine Translation (SMT) system. In this part we mainly focus on the extraction of the external features, given by toolkits which are not part of the SMT system.

The TreeTagger toolkit [Schmid, 1995] is involved in the extraction of the following features: "Proper Names", "Unknown Stems" and "Source/Target Stem". GIZA++ [Och and Ney, 2003] helps us to extract the context alignment features for POS, Word and Stems. To compute the features "Longest Target *N*-gram Length" and "Longest Source *N*-gram Length" we use the SRILM toolkit [Stolcke, 2002]. The word's constituent label ("Constituent Label") and its depth in the constituent tree ("Distance to Root") are also extracted using Bonsai (for French) [Laurent et al., 2014b], [Candito et al., 2010] or Berkeley parser (for other languages) [Petrov and Klein, 2007]. To represent hierarchical structures and extract the two features, the Natural Language ToolKit (NLTK) [Bird et al., 2009] in Python is used. The BabelNet [Navigli and Ponzetto, 2012] API and DBnary API [Sérasset, 2014] are used to extract the feature "Target polysemy count".

Finally, the features "Occurences in Google Translate" and "Occurences in Bing Translator" are extracted by using the *Google Translate* and *Bing Translator* API, respectively.

4.3.5.6 Training / Decoding Phase

Once the final feature extraction stage has been completed, we use Conditionnal Random Fields (CRF) as machine learning technique through the Wapiti toolkit [Lavergne et al., 2010].

The classifier uses all the chosen features and it is trained on a preliminary labelled French–English corpus (see next section for example of corpora directly usable with our toolkit). During decoding phase, the classifier determines, from a test corpus, whether a word should be labelled as "correct" or "incorrect" (respectively *Good* or *Bad*).

4.3.5.7 Adaptation to a New Language Pair

To evaluate our toolkit on another language pair (English–Spanish), we used the official data from WMT 2014 shared task on WCE.

One of the strength of our toolkit is the easiness to adapt it to another language pair within the (so-far) supported languages which are French, English, and Spanish. Thus, a few configuration parameters were changed to move from the French–English (*fr–en*) to English–Spanish (*en–es*), which are mainly the source language, the target language, and paths associated to input files.

Consequently, our WCE toolkit process *en–es* task in the same way as for *fr–en* task, but some features may not be extracted due to language-pair specificities: unavailable tools, no *N*-best, *etc.* For instance, for the *en–es* task, since the *N*-best list is not available, we cannot extract the five following internal features: "WPP Exact", "WPP Any", "Nodes", "WPP Min" and "WPP Max".

4.3.5.8 Integrating Other Toolkits: NLTK, YAML, NumPy, Scikit-learn, Pandas, Matplotlib, GIZA++, SRILM, Terp-A, TreeTagger, Berkeley Parser, bonsai-v3.2, BabelNet, DBnary, Wapiti, bonzaiboost

Our open-source *LIG-WCE Toolkit* is developed in *Python 3* and integrated several efficient existing libraries / toolkits as follows:

- Natural Language ToolKit (NLTK)[9] [Bird et al., 2009]: to represent hierarchical structures and to have various text processing libraries such as tokenization, stemming, tagging, parsing, *etc.*

[9] `http://www.nltk.org/`

- YAML[10]: to control the parameters configuration.
- Scikit-learn[11], NumPy[12], Pandas[13], Matplotlib[14]: these efficient libraries are used to some tasks such as pre-processing, cross-validation, scientific computing, data analysis and visualization.
- GIZA++ [Och and Ney, 2003]: to extract the context alignment features for POS, Word and Stems.
- SRILM toolkit [Stolcke, 2002]: to extract the features corresponding to Language Model.
- TERp-A toolkit [Snover et al., 2008]: to annotate automatically the errors with binary word-level labels by comparing hypotheses and given references.
- TreeTagger toolkit [Schmid, 1995]: to annotate the tokens with POS and lemma information.
- Bonsai[15] (for French) [Laurent et al., 2014b] [Candito et al., 2010] or Berkeley Parser[16] (for other languages) [Petrov and Klein, 2007]: to parse the tree containing syntactic annotations.
- The BabelNet[17] [Navigli and Ponzetto, 2012] API and DBnary API[18] [Sérasset, 2014]: to extract the features relating to the semantic information.
- Wapiti[19] [Lafferty et al., 2001]: to implement the Conditional Random Fields algorithm.
- bonzaiboost[20] [Laurent et al., 2014a]: to implement the boosting algorithm *Adaboost.MH* (over deeper trees).

[10] `http://pyyaml.org/`
[11] `http://scikit-learn.org`
[12] `http://www.numpy.org/`
[13] `http://pandas.pydata.org/`
[14] `https://matplotlib.org/`
[15] `https://alpage.inria.fr/statgram/frdep/fr_stat_dep_bky.html`
[16] `https://github.com/slavpetrov/berkeleyparser`
[17] `http://babelnet.org/`
[18] `http://kaiko.getalp.org/about-dbnary/`
[19] `https://wapiti.limsi.fr/`
[20] `http://bonzaiboost.gforge.inria.fr/`

4.4 Preliminary Results Using Only MT or Only ASR Features

In a preliminary experiment, we will evaluate these features for quality assessment in ASR only or MT only. In WCE_{ASR} task, two different classifiers will be used: a variant of boosting classification algorithm called *bonzaiboost* [Laurent et al., 2014a] (implementing the boosting algorithm *Adaboost.MH* over deeper trees) and the Conditional Random Fields [Lafferty et al., 2001].

We first report in Table 4.12 the baseline WCE results obtained using MT or ASR features separately. In short, we evaluate the performance of 4 WCE systems for different tasks:

- The first and second systems (WCE for ASR / ASR feat.) use ASR features described in Section 4.3.3 with two different classifiers (CRF or Boosting).
- The third system (WCE for SLT / MT feat.) uses only MT features described in Section 4.3.4 with CRF classifier.
- The fourth system (WCE for SLT / ASR feat.) uses only ASR features described in Section 4.3.3 with CRF classifier. The information of word-based alignment between f_{hyp} and e_{hyp} is used to generate WCE scores for both ASR and SLT hypothesis.

In all experiments reported in this paper, we evaluate the performance of our classifiers by using the average between the F-measure for *good* labels and the F-measure for *bad* labels that are calculated by the common evaluation metrics: Precision, Recall and F-measure for *good*/*bad* labels. Since two ASR systems are available, *F-mes1* is obtained for SLT based on *ASR*1 whereas *F-mes2* is obtained for SLT based on *ASR*2. For the results of Table 4.12, the classifier is evaluated on the *tst* part of our corpus and trained on the *dev* part.

Concerning WCE for ASR, we observe that F-measure decreases when ASR WER is lower (*F-mes2*<*F-mes1* while $WER_{ASR2} < WER_{ASR1}$). So quality assessment in ASR

task feat. type	**WCE for ASR** ASR feat. $p(q\|x_f, f)$ (CRFs)	**WCE for ASR** ASR feat. $p(q\|x_f, f)$ (Boosting)	**WCE for SLT** MT feat. $p(q\|f, e)$	**WCE for SLT** ASR feat. $p_{ASR}(q\|x_f, f)$ projected to e
F-mes1	68.71%	64.27%	64.69%*	53.85%
F-mes2	59.83%	62.61%	64.48%*	48.67%

Table 4.12: WCE performance with different feature sets for *tst* set (training is made on *dev* set) - *for MT feat, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features lead to 63.09% and 62.33% for *F-mes1* and *F-mes2*, respectively.

seems to become harder as the ASR system improves. This could be due to the fact that the ASR1 errors recovered by bigger LM in ASR2 system were easier to detect. Anyway, this conclusion should be considered with caution since both results (*F-mes1* and *F-mes2*) are not directly comparable because they are evaluated on different references (proportion of *good*/*bad* labels differ as ASR system differ). The effect of the classifier (CRF or Boosting) is not conclusive since CRF is better for *F-mes1* and worse for *F-mes2*. Anyway, we decide to use CRF for all our future experiments since this is the classifier integrated in our WCE-LIG toolkit [Servan et al., 2015].

To assess WCE for SLT, the observed F-measure is better using MT features rather than ASR features (quality assessment for SLT more dependent of MT features than ASR features). Again, F-measure decreases when ASR WER is lower (*F-mes2*<*F-mes1* while $WER_{ASR2} < WER_{ASR1}$). For MT features, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features lead to 63.09% and 62.33% for *F-mes1* and *F-mes2* respectively.

4.5 Conclusion

In this chapter, we introduced a new quality assessment task: word confidence estimation (WCE) for spoken language translation (SLT) with the following contributions:

- A specific corpus, distributed to the research community[21] was built for this purpose.

- We formalized WCE for SLT and proposed several approaches based on several types of features: Machine Translation (MT) based features, Automatic Speech Recognition (ASR) based features, as well as combined or joint features using ASR and MT information that will be detailed in the next chapter.

- For reproducible research, most features [22] and algorithms used in this paper are available through our toolkit called *LIG-WCE Toolkit*. This package is made available on a *GitHub* repository[23] under the licence GPL V3.

- The preliminary results on quality assessment were based on two separate WCE classifiers (one for quality assessment in ASR and one for quality assessment in MT).

- We also experiment with two ASR systems that have different performances in order to analyze the behaviors of our SLT quality assessment algorithms at different levels of word error rate (WER).

In the next chapter, we will propose a unique *joint* model based on different feature types (ASR and MT features). It is noticeable that we will propose and compare *combined features* model versus *joint features* model. We will further operate feature selection using this *joint* model and analyzing which features (from ASR or MT) are the most prominent for quality assessment in speech translation.

[21] `https://github.com/besacier/WCE-SLT-LIG`
[22] MT features already available, ASR features available soon.
[23] `https://github.com/besacier/WCE-LIG`

Chapter 5

Joint ASR and MT Features for Confidence Estimation

5.1 Combined Features versus Joint Features

5.1.1 Motivation

In the previous chapter, we described two strategies to assess WCE system for SLT using either ASR features or MT features and analysed the preliminary results on quality assessment of two separate WCE classifiers applying two ASR systems. However, we might not investigate the impact of both ASR features and MT features on the performance of WCE system for SLT.

Therefore, this chapter begins by presenting two proposed methods using SLT features (both ASR features and MT features), namely a unique *joint features* model and a *combined features* model. It will then go on to operate feature selection strategy using *joint features* model and analyse which features (from ASR or MT) are the most prominent for quality assessment in speech translation[1]. Note that we will reuse the experimental settings presented in Section 4.2 for the experiments of this chapter.

[1]Many findings in this chapter were published in [Le et al., 2016a].

5.1.2 Proposed Methods

We now report in Table 5.2 WCE for SLT results obtained using both MT and ASR features. More precisely we evaluate two different approaches (*combination* and *joint*):

- The first system (WCE for SLT / MT+ASR feat.) combines the output of two separate classifiers based on ASR and MT features. In this approach, ASR-based confidence score of source side is projected to target SLT output and combined with MT-based confidence score as shown in *Equation 6* (we did not tune the coefficient α and we set it to 0.5).

- The second system (joint feat.) trains a single WCE system for SLT (evaluating $p(q|x_f, f, e)$ as in *Equation 4.1* using joint ASR features and MT features. All ASR features are projected to the target words using automatic word alignments. However, a problem occurs when a target word does not have any source word aligned to it. In this case, we decide to duplicate the ASR features of its previous target word. Another problem occurs when a target word is aligned to more than one source word. In that case, there are several strategies to infer the 9 ASR features: average or max over numerical values, selection or concatenation over symbolic values (for F-word and F-POS), *etc.* Three different variants of these strategies (shown in Table 5.1) are evaluated here.

***ASR* Feat**	**Joint 1**	**Joint 2**	**Joint 3**
F-post	avg(F-post1, F-post2)	avg(F-post1, F-post2)	avg(F-post1, F-post2)
F-log	avg(F-log1, F-log2)	avg(F-log1, F-log2)	avg(F-log1, F-log2)
F-back	avg(F-back1, F-back2)	avg(F-back1, F-back2)	avg(F-back1, F-back2)
F-dur	max(F-dur1, F-dur2)	max(F-dur1, F-dur2)	max(F-dur1, F-dur2)
F-3g	max(F-3g1, F-3g2)	max(F-3g1, F-3g2)	max(F-3g1, F-3g2)
F-alt	max(F-alt1, F-alt2)	max(F-alt1, F-alt2)	max(F-alt1, F-alt2)
F-word	F-word1	F-word2	F-word1_F-word2
F-POS	F-POS1	F-POS2	F-POS1_F-POS2
F-context	F-context*	F-context*	F-context*

Table 5.1: Different strategies to project ASR features to a target word when it is aligned to more than one source word. *It should be noted that **F-context** features are the combinations of the source word (**F-word**) and one POS of source word (**F-POS**) before or one POS of source word (**F-POS**) after.

5.1.3 Results and Analysis

The results of Table 5.2 show that joint ASR and MT features only slightly improves WCE performance: *F-mes1* is slightly better than one of Table 4.12 (WCE for SLT / MT features only).

task feat. type	**WCE for SLT** MT+ASR feat. $p_{ASR}(q\|x_f,f)^{\alpha}$ $*p_{MT}(q\|e,f)^{1-\alpha}$	**WCE for SLT** Joint feat. 1 $p(q\|x_f,f,e)$	**WCE for SLT** Joint feat. 2 $p(q\|x_f,f,e)$	**WCE for SLT** Joint feat. 3 $p(q\|x_f,f,e)$
F-mes1	58.07%	64.90%*	64.84%	64.86%
F-mes2	53.66%	64.17%*	64.11%	63.87%

Table 5.2: WCE performance with combination (MT+ASR) or joint (MT, ASR) feature sets for *tst* set (training is made on *dev* set) - * For *Joint 1* feat, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features lead to 63.31% and 62.16% for *F-mes1* and *F-mes2*, respectively.

We also observe that simple combination (MT+ASR) degrades the WCE performance. This latter observation may be due to different behaviors of WCE_{MT} and WCE_{ASR} classifiers which makes the weighted combination ineffective. The relatively disappointing performance of our joint classifier may be due to an insufficient training set (only 2643 utterances in *dev*!). Finally, removing *OccurInGoogleTranslate* and *OccurInBingTranslate* features for *Joint* lowered *F-mes* between 1% and 2%.

These observations lead us to investigate the behaviour of our WCE approaches for a large range of *good*/*bad* decision threshold.

While the previous tables provided WCE performance for a single point of interest (*good*/*bad* decision threshold set to 0.5), the curves of figures 5.1 and 5.2 show the full picture of our WCE systems (for SLT) using speech transcriptions systems *ASR*1 and *ASR*2, respectively. We observe that the classifier based on ASR features has a very different behaviour than the classifier based on MT features which explains why their simple combination (MT+ASR) does not work very well for the default decision threshold (0.5). However, for threshold above 0.75, the use of joint ASR and MT features is slightly beneficial compared to MT features only. This is interesting because higher thresholds improves the F-measure on *bad* labels (so improves error detection).

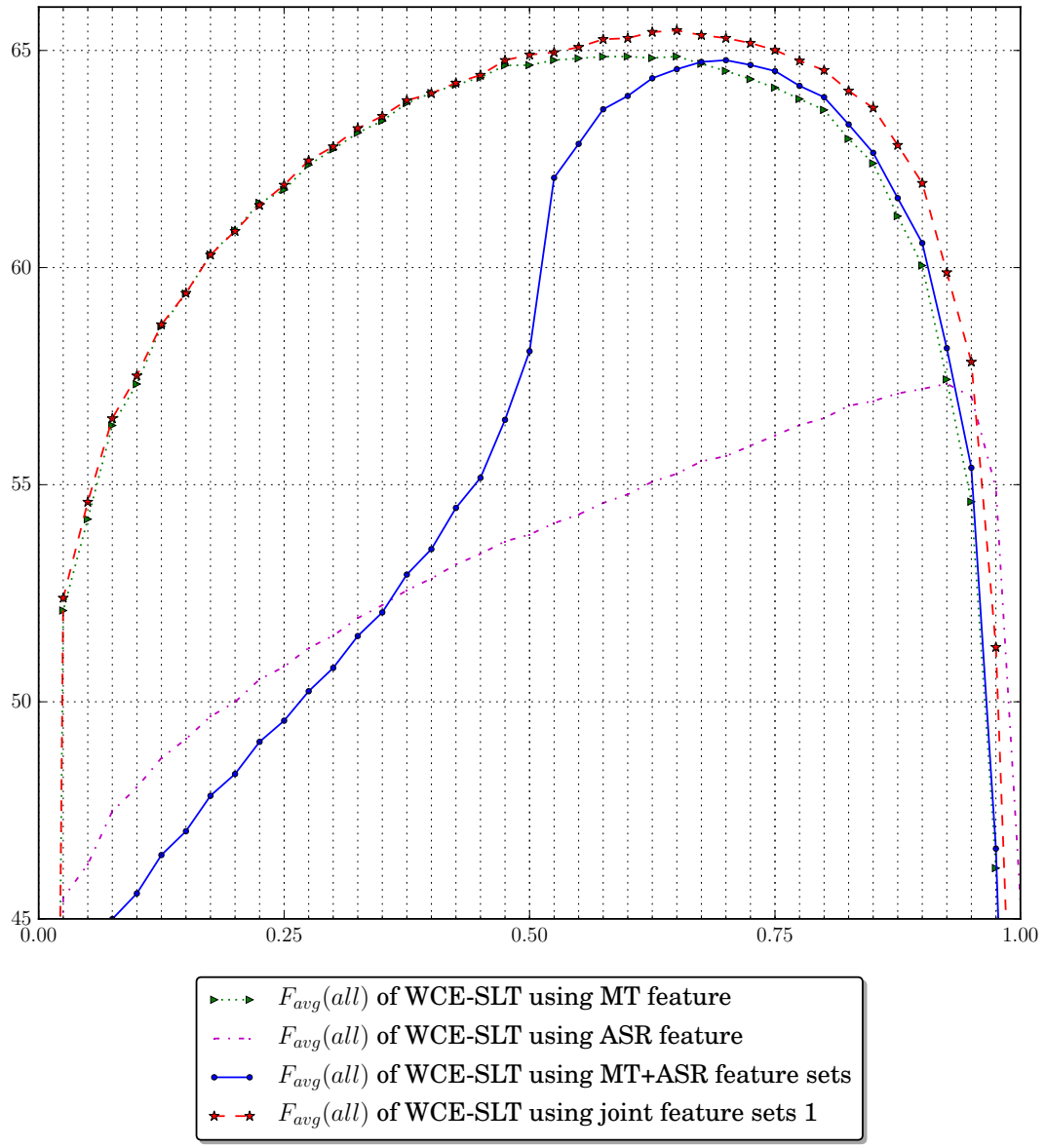

Figure 5.1: Evolution of system performance (y-axis - *F-mes1* - ASR1) for *tst* corpus (4050 utt) along decision threshold variation (x-axis) - training is made on *dev* corpus (2643 utt).

Both curves are similar whatever the ASR system used. These results suggest that with enough development data for appropriate threshold tuning (which we do not have for this very new task), the use of both ASR and MT features should improve error detection in speech translation (blue and red curves are above the green curve for higher decision threshold[2]). We also analyzed the F-measure curves for *bad* and *good* labels separately: if we consider, for instance *ASR*1 system, for decision threshold equals to 0.75, the F-measure on *bad* labels is equivalent (52%) for 3 systems (*Joint, MT+ASR*

[2]Corresponding to optimization of the F-measure on *bad* labels (errors).

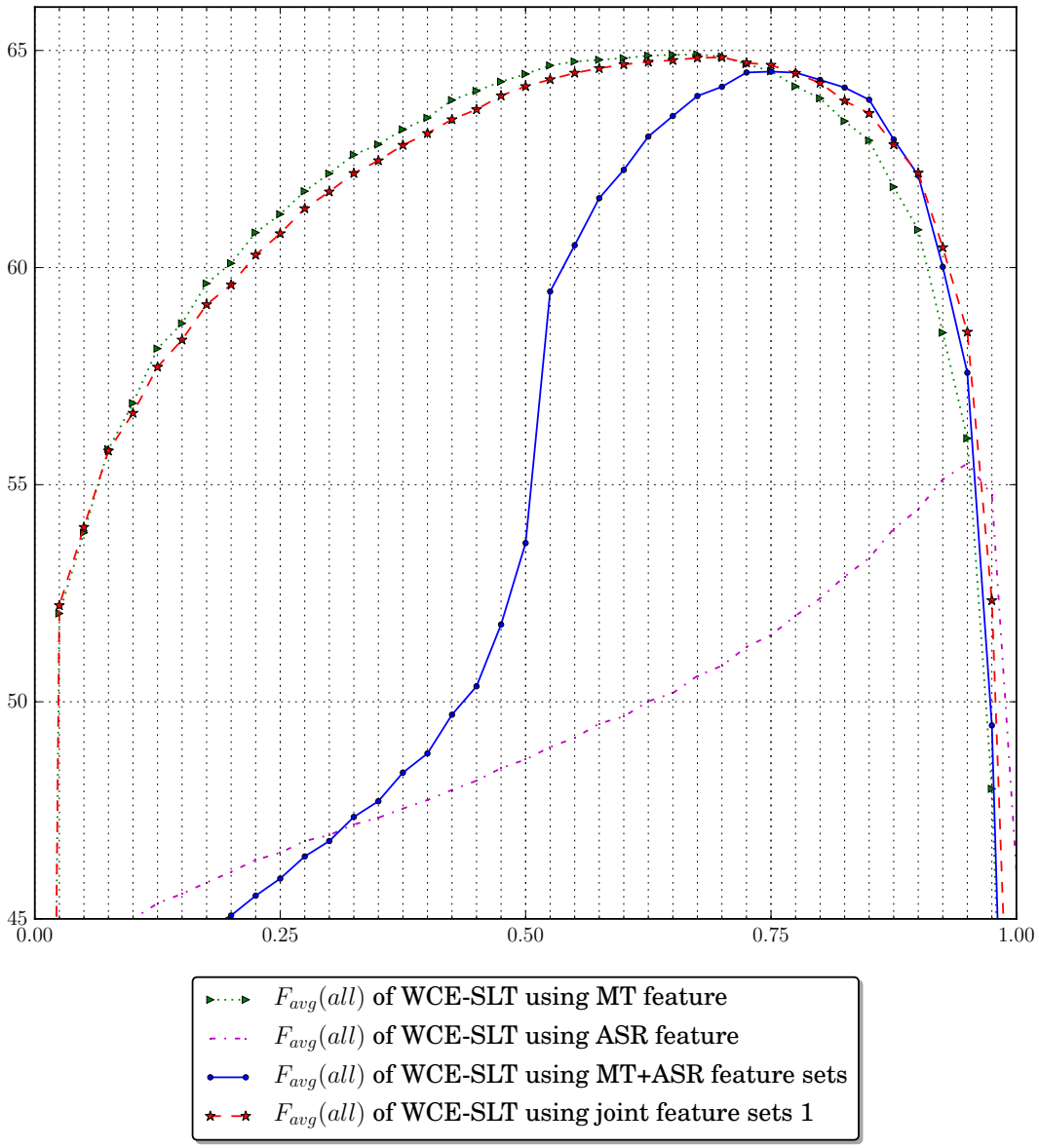

Figure 5.2: Evolution of system performance (y-axis - *F-mes2* - ASR2) for *tst* corpus (4050 utt) along decision threshold variation (x-axis) - training is made on *dev* corpus (2643 utt).

and *MT*) while the F-measure on *good* labels is 76% when using *MT* features only, 78% when using *Joint* features and 77% when using *MT+ASR* features. In other words, for a fixed performance on *bad* labels, the F-measure on *good* labels is improved using all information available (ASR and MT features). Finally, if we focus on *Joint* versus *MT+ASR*, we notice that the range of the threshold where performance are stable is larger for *Joint* than for *MT+ASR*.

5.2 Feature Selection

5.2.1 Motivation

As discussed in the above section, we could see that WCE performances using *joint* classifier given different SLT systems are dependent on their *good/bad* decision thresholds.

Therefore, in this section, we try to better understand the contribution of each (ASR or MT) feature by applying feature selection on our *joint* WCE classifier. In these experiments, we decide to keep two prominent MT features (*OccurInGoogleTranslate*, *OccurInBingTranslate* features) and the default decision threshold (0.5).

5.2.2 Proposed Methods

We choose the Sequential Backward Selection (SBS) algorithm which is a top-down algorithm starting from a feature set noted Y_k (which denotes the set of all features) and sequentially removing the most irrelevant one (x) that maximizes the Mean F-Measure, $MF(Y_k - x)$. In our work, we examine until the set Y_k contains only one remaining feature. Algorithm 2 summarizes the whole process.

Algorithm 2 Sequential Backward Selection (SBS) algorithm for feature selection. Y_k denotes the set of all features and x is the feature removed at each step of the algorithm.

```
while size of Y_k > 0 do
    maxval = 0
    for x ∈ Y_k do
        if maxval < MF(Y_k − x) then
            maxval ← MF(Y_k − x)
            worstfeat ← x
        end if
    end for
    remove worstfeat from Y_k
end while
```

5.2.3 Results and Analysis

The results of the SBS algorithm can be found in Table 5.3 which ranks all joint features used in WCE for SLT by order of importance after applying the algorithm on *dev*. We can see that the SBS algorithm is not very stable and is clearly influenced by the ASR system (*ASR*1 or *ASR*2) considered in SLT. Anyway, if we focus on the features that are in the top-10 best in both cases, we find that the most relevant ones are:

- *Alignment Features* (source and target collocations features)
- *Occur in Google Translate* and *Occur in Bing Translate* (diagnostic from other MT systems),
- *Longest Source N-gram Length, Target Backoff Behaviour* (source or target *N*-gram features),
- *Word Posterior Probability Max (WPP Max)* (graph topology feature)

Rank ***ASR1***	**Rank** ***ASR2***	**Feature**	**Rank** ***ASR1***	**Rank** ***ASR2***	**Feature**
1	1	Alignment Features	18	20	Unknown Stem
2	2	Occur in Bing Translate	19	29	Number of Word Occurrences
3	4	Longest Source *N*-gram Length	20	28	Polysemy Count - Target
4	3	WPP Max	21	19	**F-dur**
5	6	Occur in Google Translate	22	12	Punctuation
6	24	**F-back**	23	21	Constituent Label
7	11	**F-context**	24	25	**F-word**
8	27	**F-alt**	25	23	Longest Target *N*-gram Length
9	7	Target Backoff Behaviour	26	10	POS Context Alignment
10	5	Word Context Alignment	27	26	WPP Exact
11	30	Stem Context Alignment	28	18	WPP Any
12	31	Numeric	29	22	Proper Name
13	13	Distance to Root	30	8	Number of Stem Occurrences
14	9	**F-3g**	31	16	**F-POS**
15	17	Stop Word	32	33	**F-post**
16	15	Nodes	33	32	**F-log**
17	14	WPP Min			

Table 5.3: Rank of each feature according to the Sequential Backward Selection algorithm - WCE for SLT task - Joint (ASR,MT) features used - Feature selection applied to *dev* corpus for both *ASR1* and *ASR2* - ASR features are in bold.

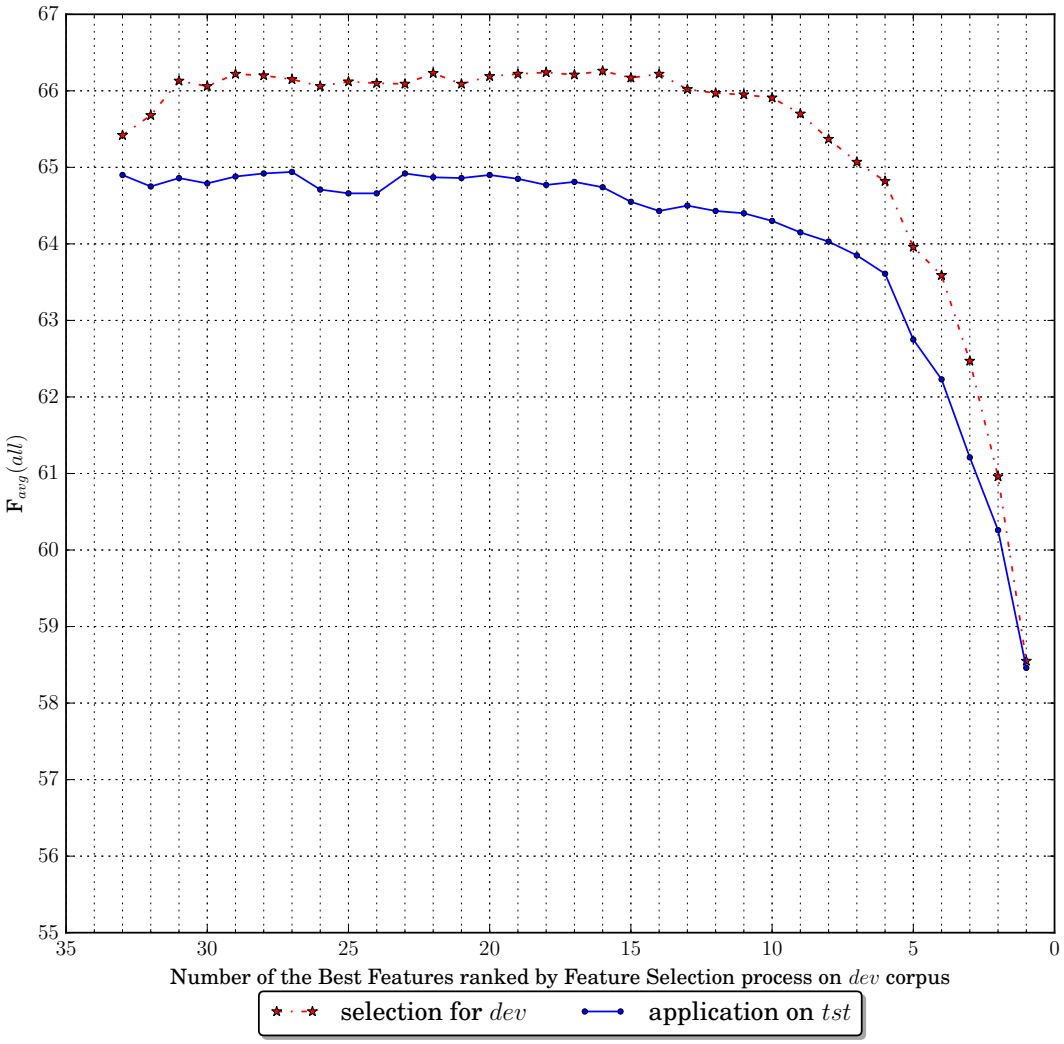

Figure 5.3: Evolution of WCE performance for *dev* (features selected) and *tst* corpora when feature selection using SBS algorithm is made on *dev* (*ASR*1 system).

We also observe that the most relevant ASR features (in bold in Table 5.3) are *F-back*, *F-3g* and *F-context* (linguistic and context features) whereas ASR lexical, acoustic and graph based features are among the worst (*F-POS, F-dur* and *F-post*). So, in our experimental setting, it seems that MT features are more influent than ASR features. Interestingly, "source and target collocations features" (*Alignment Features*) and *"Occur in Bing Translate"* are the most prominent features (rank 1 and rank 2, respectively) when applied to *dev* corpus for both *ASR1* and *ASR2*. Besides, the graph topology feature extracted from a confusion network *WPP Max* outperforms the others such as *Nodes* and *WPP Min*. Nevertheless, two other features including *WPP Exact* and *WPP any* are proven to be weak in accordance with their bottom-most positions against the two above

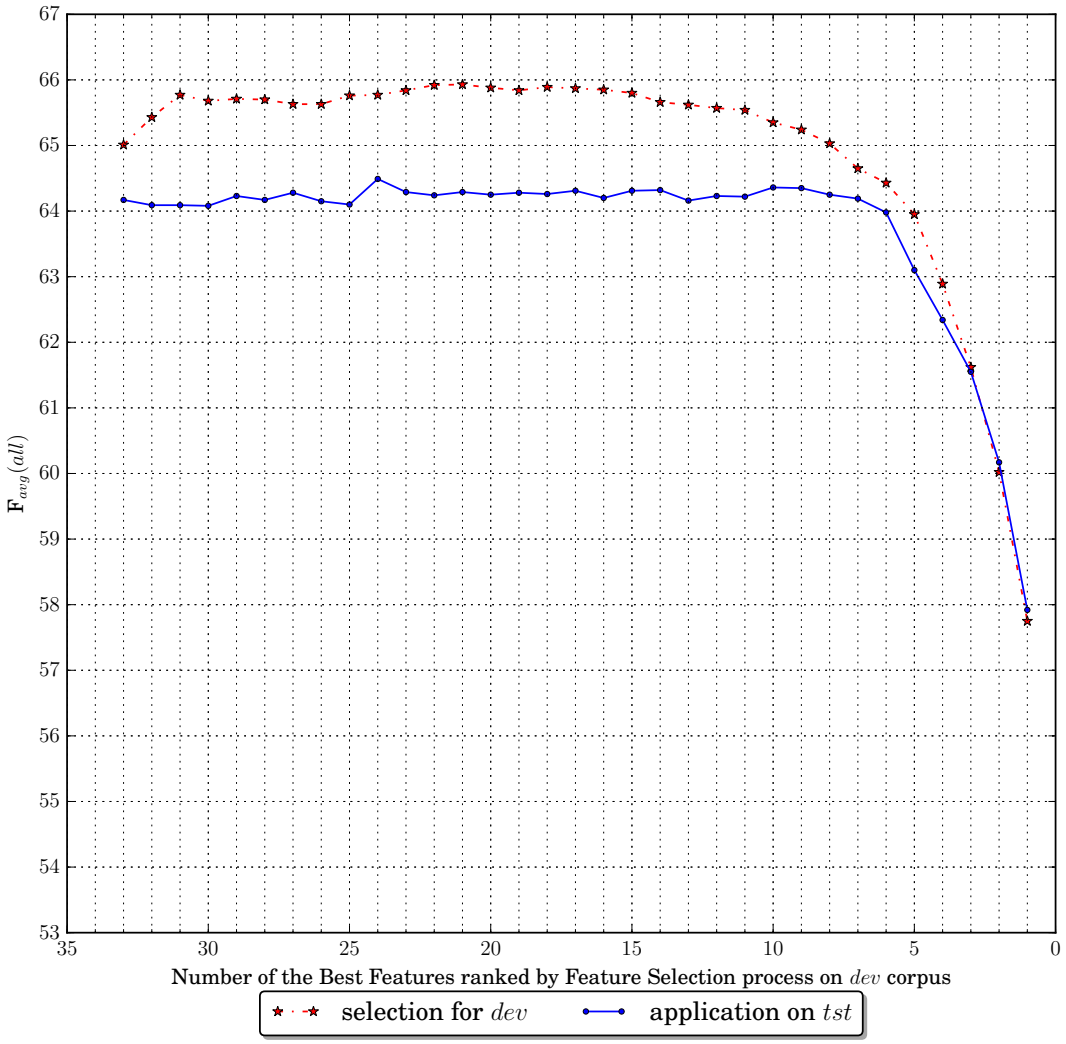

Figure 5.4: Evolution of WCE performance for *dev* (features selected) and *tst* corpora when feature selection using SBS algorithm is made on *dev* (*ASR*2 system).

systems whereas we were expecting to see them among the top features (as shown in [Luong et al., 2015] where *WPP Any* is among the best features for WCE in MT).

Figure 5.3 and Figure 5.4 present the evolution of WCE performance for *dev* and *tst* corpora when feature selection using SBS algorithm is made on *dev*, for *ASR*1 and *ASR*2 systems, respectively. In other words, for these two figures, we apply our SBS algorithm on *dev* which means that feature selection is done on *dev* with classifiers trained on *tst*. After that, the best feature subsets (using 33, 32, 31 until 1 feature only) are applied to *tst* corpus (with classifiers trained on *dev*)[3].

[3] 3 data sets would have been needed to (a) train classifiers, (b) apply feature selection, (c) evaluate WCE performance. Since we only have a *dev* and a *tst* set, we found this procedure acceptable.

On both figures, we observe that half of the features only contribute to the WCE process since best performances are observed with 15 to 25 features only. We also notice that optimal WCE performance is not necessarily obtained with the full feature set but it can be obtained with a subset of it.

5.3 Conclusion

In this chapter, we proposed a unique *joint* model based on different feature types (ASR and MT features). Note that we proposed and analyzed *combined features* model versus *joint features* model. In addition, we operated feature selection using this *joint* model and analyzing which features (from ASR or MT) are the most important for quality assessment in speech translation.

The proposition of a unique *joint* classifier based on different feature types (ASR and MT features) allowed us to operate feature selection and analyze which features (from ASR or MT) are the most efficient for quality assessment in speech translation. Our experiments have shown that MT features remain the most influential while ASR features can bring interesting complementary information. In all our experiments, we systematically evaluated with two ASR systems that have different performance in order to analyze the behavior of our quality assessment algorithms at different levels of word error rate (WER). This allowed us to observe that WCE performance decreases as ASR system improves.

In the next chapter, we will propose to disentangle ASR and MT errors and recast WCE for SLT as a 3-label setting problem.

Chapter 6

Disentangling ASR and MT Errors in Speech Translation

6.1 Motivation

In Chapter 4 and Chapter 5, we proposed and analysed various SLT quality assessment approaches based on word-level. Those classifiers assessed a 2-class (*good*/*bad*) problem. So, we might not identify the dominant error which is due to transcription (ASR) or to translation (MT) modules.

Therefore, this chapter addresses a relatively new quality assessment task: error detection in spoken language translation (SLT) using both automatic speech recognition (ASR) features and machine translation (MT) features. Its goal is also to extend error detection to a 3-class problem ($good/bad_{ASR}/bad_{MT}$) where we try to find the source of the SLT errors. Moreover, the 3-class problem necessitates to disentangle ASR and MT errors in the speech translation output and we propose two label extraction methods for this non trivial step. This enables - as a by-product - qualitative analysis on the SLT errors and their origin (are they due to transcription or to translation step?) on our large in-house corpus for French-to-English speech translation.

The outline of this chapter goes simply as follows: section 6.2 presents our experimental setup. Section 6.3 proposes two methods to disentangle ASR and MT errors in SLT

output. Section 6.4 describes the example with 3-label setting and Section 6.5 presents the statistics on a large French-English corpus. Section 6.6 gives some qualitative analysis of SLT errors. Section 6.7 presents our 2-class and 3-class error detection results while Section 6.8 concludes this work and gives some perspectives[1].

6.2 Dataset, ASR and MT Modules

The experimental settings contain the same configuration as in Chapter 4 and Chapter 5. We just recall them briefly here.

6.2.1 Dataset

In this chapter, we use our in-house corpus made available on a *github* repository[2] for reproductibility. The *dev* set and *tst* set of this corpus were recorded by french native speakers. Each sentence was uttered by 3 speakers, leading to 2643 and 4050 speech recordings for *dev* set and *tst* set, respectively. For each speech utterance, a quintuplet containing: ASR output (f_{hyp}), verbatim transcript (f_{ref}), text translation output ($e_{hyp_{mt}}$), speech translation output ($e_{hyp_{slt}}$) and post-edition of translation (e_{ref}) is available. The total length of the union of *dev* and *tst* is 16h52 (42 speakers - 5h51 for *dev* and 11h01 for *tst*).

6.2.2 ASR and MT Systems

To obtain the speech transcripts (f_{hyp}), we built a French ASR system based on KALDI toolkit [Povey et al., 2011b]. Acoustic models are trained using several corpora (ESTER, REPERE, ETAPE and BREF120) representing more than 600 hours of french transcribed speech. We use two 3-gram language models trained on French ESTER

[1] Most of our key findings in this chapter were published in [Le et al., 2017].
[2] `https://github.com/besacier/WCE-SLT-LIG/`

corpus [Galliano et al., 2006] as well as on French Gigaword (vocabulary size are respectively 62k and 95k). ASR systems LM weight parameters are tuned through WER on *dev* corpus. *Table 6.1* presents the performances obtained by both ASR systems.

In addition, we used *moses* phrase-based translation toolkit [Koehn et al., 2007] to translate French ASR into English (e_{hyp}). This medium-size system was trained using a subset of data provided for IWSLT 2012 evaluation [Federico et al., 2012]: Europarl, Ted and News-Commentary corpora. The total amount is about 60M words. We used an adapted target language model trained on specific data (News Crawled corpora) similar to our evaluation corpus (see [Potet et al., 2010]).

6.2.3 Obtaining Error Labels for SLT

To infer the quality (G, B) labels of our speech translation output $e_{hyp_{slt}}$, we use TERp-A toolkit [Snover et al., 2008] between $e_{hyp_{slt}}$ and e_{ref} (more details can be found in our former paper [Besacier et al., 2015]). Table 6.1 summarizes baseline ASR, MT and SLT performances obtained on our corpora, as well as the distribution of the binary labels (*good*, *bad*) inferred for both tasks.

Task	***ASR (WER)***		***MT (BLEU)***		***% G (good))***		***% B (bad)***	
	dev set	*tst* set	*dev* set	*tst* set	*dev* set	*tst* set	*dev* set	*tst* set
MT			49.13%	57.87%	76.93%	81.58%	23.07%	18.42%
SLT (ASR1)	21.86%	17.37%	26.73%	36.21%	62.03%	70.59%	37.97%	29.41%
SLT (ASR2)	16.90%	12.50%	28.89%	38.97%	63.87%	72.61%	36.13%	27.39%

Table 6.1: ASR, MT and SLT performances on our *dev* and *tst* set.

6.3 Disentangling ASR and MT Errors

In previous chapter, we only extract *good*/*bad* labels from the SLT output while it might be interesting to move from a 2-class problem to a 3-class problem in order to label our SLT hypotheses with one of the 3 following labels: *good* (*G*), *asr-error* (*B_ASR*) and *mt-error* (*B_MT*). Before training automatic systems for error detection, we need to set

such 3-class labels on our *dev* and *test* corpora. For that, we propose, in the next subsections, two slightly different methods to extract them. The first one is based on word alignments between SLT and MT and the second one is based on a simpler SLT-MT error subtraction.

6.3.1 Method 1 - Word Alignments between MT and SLT

From this simple definition, we derive our first way (*Method 1*) to generate 3-class annotations.

Let $\hat{e}_{slt} = (e_1, e_2, \ldots, e_n)$: the set of SLT hypotheses ($e_{hyp_{slt}}$); e_{k_j} denotes the j^{th} word in the sentence e_k, where $1 \leq k \leq n$

Let $\hat{e}_{mt} = (e'_1, e'_2, \ldots, e'_m)$: the set of MT hypotheses ($e_{hyp_{mt}}$); e'_{k_i} denotes the i^{th} word in the sentence e'_k, where $1 \leq k \leq n$

Let $L = (l_1, l_2, \ldots, l_n)$: the set of the word alignments from sentences in $e_{hyp_{slt}}$ to related sentences in $e_{hyp_{mt}}$, where l_k contains the word alignments from sentence e_k to relevant sentence e'_k, $1 \leq k \leq n$; (e_{k_j}, e'_{k_i}) = *True*, if there is one word alignment between e_{k_j} and e'_{k_i}; (e_{k_j}, e'_{k_i}) = *False*, otherwise.

Our algorithm for *Method 1* is defined as *Algorithm 3*. This method relies on word alignments and uses MT labels. We also propose a simpler method in the next section.

6.3.2 Method 2 - Subtraction between SLT and MT Errors

Our second way to extract 3-class labels (*Method 2*) focuses on the differences between SLT hypothesis ($e_{hyp_{slt}}$) and MT hypothesis ($e_{hyp_{mt}}$). We call it *subtraction between SLT and MT errors* because we simply consider that errors present in SLT and not present in MT are due to ASR. This method has a main difference with the previous one: it does not rely on the extracted labels for MT.

Algorithm 3 *Method 1* - Using word alignments between MT and SLT

list_labels_result ← *empty_list*
for each sentence $e_k \in \hat{e}_{slt}$ **do**
  list_labels_sent ← *empty_list*
  for $j \leftarrow 1$ to $NumberOfWords(e_k)$ **do**
    if $label(e_{k_j}) =$ 'G' **then**
      add 'G' to *list_labels_sent*
    else if Existed Word Alignment (e_{k_j}, e'_{k_i}) and $label(e'_{k_i})$='B' **then**
      add '*B_MT*' to *list_labels_sent*
    else
      add '*B_ASR*' to *list_labels_sent*
    end if
  end for
  add *list_labels_sent* to *list_labels_result*
end for

Our intuition is that the number of *mt-error*s estimated will be slightly lower than for *Method 1* since we first estimate the number of *asr-error*s and the rest is considered - by default - as *mt-error*s.

With the same notations of *Method 1*, but highlighting that $L = (l_1, l_2, \ldots, l_n)$ is the set of alignments through edit distance between $e_{hyp_{slt}}$ and $e_{hyp_{mt}}$, where l_{k_i} corresponds to "Insertion" (I), "Substitution" (S), "Deletion" (D) or "Exact" (E). Our algorithm for *Method 2* is defined as Algorithm 4.

Algorithm 4 *Method 2* - Subtraction between SLT and MT errors

list_labels_result ← *empty_list*
for each sentence $e_k \in \hat{e}_{slt}$ **do**
  list_labels_sent ← *empty_list*
  for $j \leftarrow 1$ to $NumberOfWords(e_k)$ **do**
    if $label(e_{k_j}) =$ 'G' **then**
      add 'G' to *list_labels_sent*
    else if $NameOfWordAlignment(l_{k_i})$ is 'I' OR 'S' **then**
      add '*B_ASR*' to *list_labels_sent*
    else
      add '*B_MT*' to *list_labels_sent*
    end if
  end for
  add *list_labels_sent* to *list_labels_result*
end for

6.4 Example with 3-label Setting

Table 6.2 gives the edit distance between a SLT and MT hypothesis while table 6.3 shows how *Method 1* and *Method 2* set 3-class labels to the SLT hypothesis. One transcript (f_{hyp}) has 1 error. This drives 3 B labels on SLT output ($e_{hyp_{slt}}$), while $e_{hyp_{mt}}$ has only 2 B labels. As can be seen in the cases of *Method 1* and *Method 2*, we respectively have (1 B_ASR, 2 B_MT) and (2 B_ASR, 1 B_MT).

$e_{hyp_{slt}}$	surgeons	in	los	angeles	it	is	said
$e_{hyp_{mt}}$	surgeons	in	los	angeles	**	have	said
edit op.	E	E	E	E	I	S	E

Table 6.2: Example of edit distance between SLT and MT.

f_{ref}	les chirurgiens	de	los	angeles	ont		dit
f_{hyp}	les chirurgiens	de	los	angeles	on		dit
labels ASR	G G	G	G	G	B		G
$e_{hyp_{mt}}$	surgeons	in	los	angeles		have	said
labels MT	G	B	G	G		B	G
$e_{hyp_{slt}}$	surgeons	in	los	angeles	it	is	said
labels SLT (2-label)	G	B	G	G	B	B	G
labels SLT (*Method 1*)	G	B_MT	G	G	B_ASR	B_MT	G
labels SLT (*Method 2*)	G	B_MT	G	G	B_ASR	B_ASR	G
e_{ref}	the surgeons	of	los	angeles			said

Table 6.3: Example of quintuplet with 2-label and 3-label.

These differences are due to slightly different algorithms for label extraction. As Table 6.3 presents, “is” (SLT hypothesis) is aligned to “have” (MT hypothesis) and “have” (MT hypothesis) is labeled by “B”. It can therefore be assumed that “is” (SLT hypothesis) should be annotated with word-level labels by B_MT according to *Method 1*. However, using *Method 2*, “is” (SLT hypothesis) could be labeled by B_ASR because the type of word alignment between “is” (SLT hypothesis) and “have” (MT hypothesis) is substitution (S), as shown in Table 6.2.

6.5 Statistics with 3-label Setting on the Whole Corpus

Table 6.4 presents the summary statistics for the distribution of *good* (G), *asr-error* (B_ASR) and *mt-error* (B_MT) labels obtained with both label extraction methods. We see that both methods give similar statistics but slightly different rates of B_ASR and B_MT.

Task - *ASR1*	***dev* set**			***tst* set**		
	%G	***%B_ASR***	***%B_MT***	**%G**	**%B_ASR**	**%B_MT**
label/m1:Method 1	*62.03*	*19.09*	*18.89*	70.59	14.50	14.91
label/m2:Method 2	*62.03*	*22.49*	*15.49*	70.59	16.62	12.79
label/same(m1, m2)	*62.03*	*18.09*	*14.49*	70.59	13.58	11.88
label/diff(m1, m2)	*0*	*1.00*	*4.40*	0	0.92	3.03
Task - *ASR2*	***dev* set**			***tst* set**		
	%G	***%B_ASR***	***%B_MT***	**%G**	**%B_ASR**	**%B_MT**
label/m1:Method 1	*63.87*	*16.89*	*19.23*	72.61	11.92	15.47
label/m2:Method 2	*63.87*	*19.78*	*16.34*	72.61	13.58	13.81
label/same(m1, m2)	*63.87*	*16.05*	*15.50*	72.61	11.12	13.01
label/diff(m1, m2)	*0*	*0.84*	*3.73*	0	0.80	2.46

Table 6.4: Statistics with 3-label setting for *ASR1* and *ASR2*.

As can be seen from Table 6.4, it is interesting to note that while ASR system improves from *ASR1* to *ASR2*, the rate of B_ASR labels logically decreases by more than 2 points, while the rate of B_MT remains almost stable (less than 1 point difference) which makes sense since the MT system is the same in both *ASR1* and *ASR2*. These statistics show that intersection between both methods is probably a good estimation of disentangled ASR and MT errors in SLT.

6.6 Qualitative Analysis of SLT Errors

Our new 3-label setting procedure allows us to analyze the behavior of our SLT system.

We can observe sentences presented in Table 6.5 presents, as an example, few ASR and MT errors leading to many SLT errors. Indeed, this is a good way of detecting flaws in the SLT pipeline such as bad post-processing of the SLT output (numerical or text dates, for instance).

f_{ref}	peter frey est né le quatre août mille neuf cent cinquante sept à bingen
f_{hyp_1}	**pierre ferait aimé** le quatre août mille neuf cent cinquante sept à **big m**
f_{hyp_2}	**pierre** frey est né le quatre août mille neuf cent cinquante sept à **big m**
$e_{hyp_{mt}}$	peter frey was born on **4** august 1957 **to** bingen .
$e_{hyp_{slt1}}$	**pierre would liked the four** august **thousand nine hundred and fifty seven to big m**
$e_{hyp_{slt2}}$	**pierre** frey is born **the four** august **thousand nine hundred and fifty seven to big m**
e_{ref}	peter frey was born on august 4th 1957 in bingen .

Table 6.5: Example 1 - SLT hypothesis annotated with two methods - having a few *asr-errors*, a few *mt-errors* and many *slt-errors* such as 5 B_ASR1, 3 B_ASR2, 2 B_MT, 14 B_SLT1, 12 B_SLT2.

f_{ref}	malheureusement le système européen de financement gouvernemental direct est
f_{hyp_1}	malheureusement le système européen financement **gouvernementale directe et**
f_{hyp_2}	malheureusement le système européen de financement gouvernemental direct est
$e_{hyp_{mt}}$	unfortunately , the european system of direct government funding is
$e_{hyp_{slt1}}$	unfortunately the european system direct government funding
$e_{hyp_{slt2}}$	unfortunately the european system of direct government funding is
e_{ref}	unfortunately , the european system of direct government funding is
f_{ref}	victime de la croissance économique européenne lente et des déficits budgétaires
f_{hyp_1}	**victimes** de la croissance économique européenne **venant de déficit budgétaire**
f_{hyp_2}	victime de la croissance économique européenne **venant** des déficits budgétaires
$e_{hyp_{mt}}$	a victim of european economic growth **slow** and budget deficits .
$e_{hyp_{slt1}}$	**and** victims of european economic growth **from** budget deficit
$e_{hyp_{slt2}}$	a victim of european economic growth **from the** budget deficits
e_{ref}	a victim of slow european economic growth and budget deficits .

Table 6.6: Example 2 - SLT hypothesis annotated with two methods - having many *asr-errors*, a few *mt-errors* and a few *slt-errors* such as 8 B_ASR1, 1 B_ASR2, 1 B_MT, 2 B_SLT1, 2 B_SLT2.

As shown in Table 6.6, on the contrary, there are many ASR errors leading to few SLT errors (ASR errors with few consequences such as morphological substitutions - for instance in French: de/des, déficit/déficits, budgétaire/budgétaires).

Moreover, ASR errors as presented in Table 6.7 have different consequences on SLT quality (on a sample sentence, 2 ASR errors of system 1 and 2 lead to 14 and 9 SLT errors, respectively).

f_{ref}	nous ne comprenons pas ce qui se passe chez les jeunes pour qu' ils trouvent
f_{hyp_1}	nous ne comprenons pas **ceux** qui se passe chez les jeunes pour qu' ils trouvent
f_{hyp_2}	nous ne comprenons pas ce qui se passe chez les jeunes pour qu' **il trouve**
$e_{hyp_{mt}}$	we do not understand what is happening **among** young people for **that**
$e_{hyp_{slt1}}$	we do not understand **those who happens among** young people for **that**
$e_{hyp_{slt2}}$	we do not understand what is happening **among** young people
e_{ref}	we do not understand what is happening in young people 's mind for them
f_{ref}	amusant de maltraiter gratuitement un animal sans défense qui nous donne
f_{hyp_1}	amusant de **maltraité** gratuitement un animal sans défense qui nous
f_{hyp_2}	amusant de maltraiter gratuitement un animal sans défense qui nous donne
$e_{hyp_{mt}}$	**they are fun** to mistreat **free a** defenceless animal
$e_{hyp_{slt1}}$	**they** find **fun free** mistreated **a** defenceless animal
$e_{hyp_{slt2}}$	to find **it** amusing to mistreat **free a** defenceless animal
e_{ref}	to find amusing to mistreat defenceless animals without reason ,
f_{ref}	de l' affection de l' amitié et nous tient compagnie
f_{hyp_1}	de l' affection de l' amitié nous tient compagnie
f_{hyp_2}	de l' affection de l' amitié nous tient compagnie
$e_{hyp_{mt}}$	which gives us **the affection** , friendship and **keeps us airline** .
$e_{hyp_{slt1}}$	which **we affection of** friendship **we takes** company
$e_{hyp_{slt2}}$	which gives us **the affection of** friendship **we takes** company
e_{ref}	which gives us love , friendship and companionship .

Table 6.7: Example 3 - SLT hypothesis annotated with two methods - having the same number of *asr-errors*, but the different number of *slt-errors* extracted from *ASR1* and *ASR2* such as 2 B_ASR1, 2 B_ASR2, 12 B_MT, 14 B_SLT1, 9 B_SLT2.

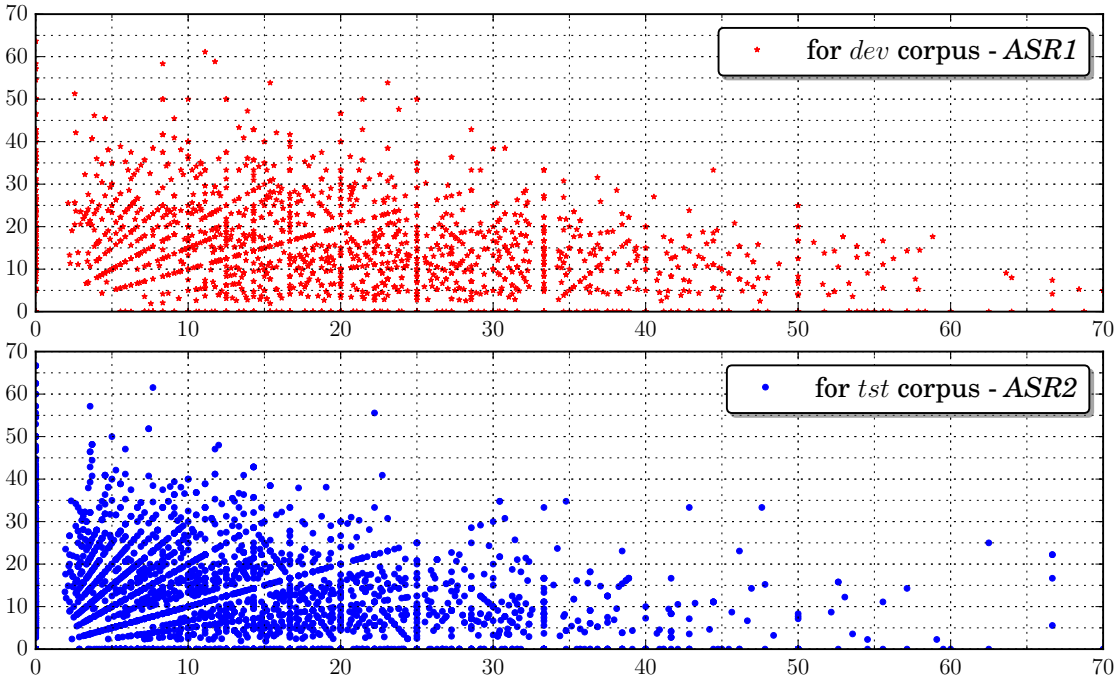

Figure 6.1: Example of the rate (%) of ASR errors (x-axis) versus (%) MT errors (y-axis) - for *dev/ASR1* and *tst/ASR2*.

In addition, Figure 6.1 shows how our speech utterances are distributed in the two-dimensional (B_{ASR}, B_{MT}) error space.

6.7 Results and Analysis

We report in Table 6.8 our first attempt to build an error detection system in SLT as a 3-class problem (*joint* approach only). We made our experiment by training and evaluating the model on *Intersection(m1, m2)* which corresponds to high confidence in the labels[3].

We compared two different approaches: *One-Step* is a single classifier for the 3-class problem while *Two-Step* first applies the 2 class (*G/B*) system and a second classifier distinguishes B_{ASR} and B_{MT} errors. Not much difference in F-measure is observed between both approaches. Table 6.9 also presents the confusion matrix between B_{ASR} and B_{MT} for the correctly detected (true) errors. Despite the relatively low F-scores of

[3]However, we observed (results not reported here) that the use of different label sets (*Method 1*, *Method 2*, *Intersection(Method 1, Method 2)* does not have a strong influence on the results.

table 6.8, we see that our 3-labels classifier obtains encouraging confusion matrices in order to automatically disentangle B_{ASR} and B_{MT} on true errors.

2-class Full Corpus			**3-class Intersection Corpus (m1, m2)**				
				One-Step		**Two-Step**	
	ASR1	***ASR2***		***ASR1***	***ASR2***	***ASR1***	***ASR2***
F_G	81.79	83.17	F_G	85.00	85.00	84.00	85.00
F_B	48.00	45.17	F_{B_ASR}	44.00	42.00	44.00	42.00
			F_{B_MT}	14.00	15.00	16.00	17.00
F_{avg}	64.90	64.17	F_{avg}	47.67	47.33	48.00	48.00

Table 6.8: Error Detection Performance (2-label vs 3-label) on SLT output for tst set (training is made on *dev* set).

(1) Ref \ Hyp	***ASR1***		***ASR2***	
	B_ASR	***B_MT***	**B_ASR**	**B_MT**
B_ASR	*85.75%*	*14.25%*	81.57%	18.43%
B_MT	*44.46%*	*55.54%*	34.53%	65.47%
(2) Ref \ Hyp	***ASR1***		***ASR2***	
	B_ASR	***B_MT***	**B_ASR**	**B_MT**
B_ASR	*83.14%*	*16.86%*	80.02%	19.98%
B_MT	*49.41%*	*50.59%*	41.49%	58.51%

Table 6.9: Confusion Matrix on Correctly Detected Errors Subset for 3-class (1) One-Step; (2) Two-Step.

6.8 Conclusion

In conclusion, we proposed two methods to disentangle ASR and MT errors in speech translation. The binary error detection problem was recast as a 3-class labeling problem (*good, asr-error, mt-error*). Firstly, two methods were proposed for the non trivial label setting and it was shown that both give consistent results. Secondly, automatic detection of error types, using joint ASR and MT features, was evaluated and encouraging results were displayed on a French-English speech translation task. We believe that such a new task (not only detecting errors but also their cause) is interesting to build better informed speech translation systems, especially in interactive speech translation use cases.

Chapter 7

Better Evaluation of ASR in Speech Translation Context Using Word Embeddings

7.1 Motivation

In spoken language translation (SLT), the ability of Word Error Rate (WER) metric to evaluate the real impact of the ASR module on the whole SLT pipeline is often questioned. This was investigated in past studies where researchers tried to propose a better evaluation of ASR in speech translation scenarios. Dixon et al. [2011] investigated how SLT performed as they changed speech decoder parameters. It was shown that sub-optimal WER values could give comparable BLEU scores at faster decoding speeds. The authors of [Bechet et al., 2015] analyzed ASR error segments that have a high negative impact on SLT performance and demonstrated that removing such segments prior to translation can improve SLT. The same year, Ruiz and Federico [2015] proposed a Phonetically-Oriented Word Error Rate (POWER) for speech recognition evaluation which incorporates the alignment of phonemes to better trace the impact of Levenshtein error types in speech recognition on downstream tasks (such as information retrieval,

spoken language understanding, speech translation, etc.). Moreover, the need to evaluate ASR when its output is used by human subjects (predict how useful that ASR output would be to humans) was also highlighted by Favre et al. [2013]. Finally, while some authors [He et al., 2011] proposed an end-to-end BLEU-oriented global optimization of ASR system parameters in order to improve translation quality, such an end-to-end optimization is not always possible in practical applications where a same ASR system is designed for several downstream uses. Thus, we believe that a better evaluation of the ASR module itself should be investigated.

This chapter rests upon the above papers as well as on the former research of [Vilar et al., 2006] who noticed that many ASR substitution errors (the most frequent type of ASR error) are due to slight morphological changes (such as plural/singular substitution), limiting the impact on SLT performance. We have also noticed this in section 6.6 of previous chapter. Thus, the current WER metric – which gives the same weight to any substitution – is probably sub-optimal for evaluating ASR module in a SLT framework. We propose a simple extension of WER in order to penalize differently substitution errors according to their context using word embeddings. For instance, the proposed metric should penalize less morphological changes that have a smaller impact on SLT. We show that the new proposed metric is better correlated with SLT performances. Oracle experiments are also conducted to show the ability of our metric to find better hypotheses (to be translated) in the ASR N-best. Finally, we propose a preliminary experiment where ASR tuning is based on our new metric. For reproducible experiments, code allowing to call our modified WER and corpora used are made available to the research community.

The rest of the chapter goes simply as follows: section 2 summarizes related works on evaluation metrics that use word embeddings. Section 3 presents our modified WER metric which allows to consider near matches in substitution errors. Section 4 details the experimental settings and section 5 presents our results. Section 6 concludes this work[1].

[1]Many of the findings described in this chapter were published in [Le et al., 2016c]. The code of the new metric was designed in collaboration with C. Servan (Post-doc at GETALP).

7.2 Word Error Rate with Embeddings (WER-E)

The Word Error Rate is the main metric applied to Automatic Speech Recognition evaluation. Its estimation is based on the Levenshtein distance, which is defined as the minimum number of editing steps needed to match an hypothesis and a reference.

7.2.1 Running Example

	un	nord	westphalie	un	d'	engagement	parmi	de	nation	souveraine
un	0	1	2	3	4	5	6	7	8	9
ordre	1	1	2	3	4	5	6	7	8	9
westphalien	2	2	2	3	4	5	6	7	8	9
d'	3	3	3	3	3	4	5	6	7	8
engagements	4	4	4	4	4	4	5	6	7	8
parmi	5	5	5	5	5	5	4	5	6	7
des	6	6	6	6	6	6	5	5	6	7
nations	7	7	7	7	7	7	6	6	6	7
souveraines	8	8	8	8	8	8	7	7	7	7
Alignment:	A	I	S	S	A	S	A	S	S	S
Cost:	0	1	1	1	0	1	0	1	1	1

Table 7.1: Example (in French) of the Word Error Rate estimation between a hypothesis (on the top) and a reference (on the left).

In table 7.1, we compare an hypothesis (on the top) and a reference (on the left): the score is defined as the lowest-cost alignment path (in grey) from the beginning of both sentences (top left corner) to the end of both sentences (on the lower-right corner). The intensity of the colour in the alignment path indicates the match level: lighter grey for matches, mid-dark grey for *substitutions* and dark grey for *insertions* and *deletions*. The score sums the number of *insertions*, *deletions* and *substitutions*. Then, this sum is normalized by the length of the reference. In our example, the *WER* is calculated as the following:

$$WER = \frac{\#Ins + \#Sub + \#Del}{\#Total\ of\ words\ in\ the\ reference} = \frac{1+6}{9} \approx 0.78 \tag{7.1}$$

7.2.2 Adding Word Embeddings

The main drawback of *WER* is that it does not gives credit to near matches. For instance, in table 7.1, the hypothesis contains the word “souveraine”, which is close to the word “souveraines” in the reference. Both are morphological variants of a same word and WER considers this difference as a *Substitution*, while their cosine distance in the continuous space of our word embeddings is only 0.43.

	un	nord	westphalie	un	d’	engagement	parmi	de	nation	souveraine
un	0	1	2	3	4	5	6	7	8	9
ordre	1	1.01	2.07	2.93	4.15	4.89	6.07	7.03	8.05	9.01
westphalien	2	1.79	1.73	2.83	3.93	5.38	5.80	6.90	7.75	8.85
d’	3	3.05	2.97	2.21	2.83	3.83	4.83	5.83	6.83	7.83
engagements	4	3.94	4.02	4.15	3.41	3.30	5.01	5.91	6.92	7.81
parmi	5	4.77	4.80	5.13	5.15	4.61	3.30	4.30	5.30	6.30
des	6	6.04	5.85	5.80	5.61	6.24	4.30	3.64	5.49	6.12
nations	7	6.87	6.83	6.77	6.85	6.55	5.30	5.26	4.42	6.43
souveraines	8	7.92	7.71	7.99	7.71	7.82	6.30	6.15	6.10	**4.85**
Alignment:	A	I	S	S	A	S	A	S	S	S
Cost:	0	1	1.07	0.75	0	0.47	0	0.35	0.78	0.43

Table 7.2: WER-E estimation with word embeddings. *Substitution* score is replaced by a cosine distance, without questionning the best alignment.

Our main idea is to find a way to include near matches in the metric without using lexico-semantic data such as Wordnet. Since word embeddings can model syntactic and semantic proximity [Mikolov et al., 2013a,c], we use them to estimate a cosine similarity between two words in a *substitution*. This cosine similarity (S_c in [-1,1]) is used to compute a cosine distance (D_c) (see equation 7.2). The *substitution* score (0 or 1) is replaced by the cosine distance between two words (continuous value in [0,2]).

$$D_c(W_1, W_2) = 1 - S_c(W_1, W_2) \tag{7.2}$$

From this, two variants of the metric are possible. Firstly, in table 7.2, we apply the *WER* alignment algorithm with classical *substitution* cost (we do not modify the alignment path of table 7.1) and we replace only the *substitution* scores by the cosine distance. We

call it "WER *with embeddings*" (*WER-E*). Secondly, in table 7.3, we propose to replace *substitution* cost by the cosine distance to compute the best alignment path. We call this last WER variant "WER *soft*" (*WER-S*). Therefore, from table 7.2 and table 7.3, we calculate *WER-E* and *WER-S* as the following:

$$\begin{aligned} \textit{WER-E} &= \frac{\textit{Cost}(\#\textit{Ins}) + \textit{Cost}(\#\textit{Sub}) + \textit{Cost}(\#\textit{Del})}{\#\textit{Total of words in the reference}} \\ &= \frac{1 + (1.07 + 0.75 + 0.47 + 0.35 + 0.78 + 0.43)}{9} \approx 0.54 \end{aligned} \tag{7.3}$$

$$\begin{aligned} \textit{WER-S} &= \frac{\textit{Cost}(\#\textit{Ins}) + \textit{Cost}(\#\textit{Sub}) + \textit{Cost}(\#\textit{Del})}{\#\textit{Total of words in the reference}} \\ &= \frac{1 + (1.01 + 0.73 + 0.47 + 0.35 + 0.78 + 0.43)}{9} = 0.53 \end{aligned} \tag{7.4}$$

In the first case (table 7.2), we can observe a *WER-E* score (0.54) lower than the classical *WER* estimation (0.78). Since we do not question the alignment path in this case, we do not obtain the lowest score possible. The second case, presented in table 7.3, enables us to get another alignment path, and thus gets the lowest score possible (0.53).

This new feature takes into account near matches between words. For instance, words "westphalie" and "westphalien" are close enough to have a low distance. In the alignment proposed in table 7.3, the alignment changed and we got a lower score.

7.3 Experimental Setup

The experimental settings contain the same configuration as in Chapter 4 and Chapter 5. For ASR output, the N-best lists (N=1000) were also generated for each utterance.

Table 7.4 gives 2 examples of SLT output obtained. Table 7.5 summarizes baseline ASR, MT and SLT performances obtained on our corpora. We score translations obtained with the following automatic metrics: TER [Snover et al., 2006], BLEU [Papineni et al., 2002] and METEOR [Banerjee and Lavie, 2005] using post-edition references (e_{ref}). Note that we used the option *lm-scale* = 10 when generating *N*-best hypotheses from ASR system ($N = 1000$) instead of applying *lm-scale* = 12 for *SLT*

	un	nord	westphalie	un	d'	engagement	parmi	de	nation	souveraine
un	0	1	2	3	4	5	6	7	8	9
ordre	1	1.01	2.01	2.93	3.93	4.89	5.89	6.89	7.89	8.89
westphalien	2	1.79	1.74	2.74	3.74	4.74	5.74	6.72	7.61	8.61
d'	3	2.79	2.74	2.21	2.74	3.74	4.74	5.74	6.74	7.74
engagements	4	3.79	3.74	3.21	3.42	3.21	4.21	5.21	6.21	7.21
parmi	5	4.77	4.65	4.21	4.21	4.21	3.21	4.21	5.21	6.21
des	6	5.77	5.65	5.21	4.68	5.21	4.21	3.55	4.55	5.55
nations	7	6.77	6.57	6.21	5.68	5.63	5.21	4.55	4.34	5.34
souveraines	8	7.77	7.57	7.21	6.68	6.63	6.21	5.55	5.34	4.76
Alignment:	A	S	S	I	A	S	A	S	S	S
Cost:	0	1.01	0.73	1	0	0.47	0	0.35	0.78	0.43

Table 7.3: WER-S estimation with word embeddings. *Substitution* score is replaced by a cosine distance **and** we recalculate the best alignment.

REF ASR	ce serait intéressant de voir un ordinateur présentant ce même système	WER	WER-E	WER-S
OptWER	ce **sera** intéressant de voir un ordinateur présentant ce même système	9.09	2.43	2.43
OptWER-E	ce **serait** intéressant de voir un ordinateur présentant ce même système	0.00	0.00	0.00
REF SLT	it would be interesting to see a computer with this same system	TER	SentBLEU	METEOR
OptWER - SLT	this **will** be interesting to see a computer with the same system	33.33	62.63	49.33
OptWER-E - SLT	it **would** be interesting to see a computer with the same system	16.67	79.11	92.73

REF ASR	en bref ils craignent que tous les sacrifices entrepris pour stabiliser les prix aient été vains	WER	WER-E	WER-S
OptWER	en bref **il craignait** que tous les sacrifices ces entreprises pour stabiliser les prix et était vingt	43.75	34.65	33.26
OptWER-E	en bref **ils craignent** que tous les sacrifices ces entreprises pour stabiliser les prix et était vingt	31.25	26.80	25.41
REF SLT	in short they fear that all the sacrifices made to stabilize prices have been fruitless	TER	SentBLEU	METEOR
OptWER - SLT	in short **it feared** that all the sacrifices these companies to stabilise prices and was 20	60.00	26.22	34.84
OptWER-E - SLT	in short **they fear** that all the sacrifices these companies to stabilise prices and was 20	46.67	50.44	40.08

Table 7.4: ASR and SLT examples (explanations given in *section 7.4.5*).

(ASR1) system presented in subsection 4.2.2 of chapter 4. Therefore, the scores *WER* for the tasks *dev* and *test* in table 7.5 and *SLT (ASR1)* in table 4.4 are nearly equal.

Tasks	*metrics*	ASR Ref.	ASR 1-best
dev	*WER*	–	21.92
	TER	38.84	55.64
	BLEU	43.05	30.81
	METEOR	40.73	34.02
test	*WER*	–	17.46
	TER	45.64	58.70
	BLEU	44.71	34.27
	METEOR	39.10	34.27

Table 7.5: Baseline ASR, MT and SLT performance on our *dev* and *test* sets - translations are scored w/o punctuation.

7.4 Results and Analysis

This section first presents the results obtained in ASR, according to our new metrics. Then, we analyze the correlation of the ASR metrics (WER, WER-E, WER-S) with SLT performances. After that, Oracle experiments are conducted to compare the ASR metrics in their ability to find (before translation) promising hypotheses in the ASR N-best. Finally, we present a preliminary experiment to tune ASR output based on our proposed metric. For all the experiments, the MT system never changes and is the one described in section 4.2.3 of chapter 4.

Tasks	*metrics*	ASR	Oracle from N-best		
		1-best	WER	WER-E	WER-S
dev	*WER*	21.92	12.01	12.16	12.15
	WER-E	18.10	10.45	9.98	10.04
	WER-S	17.41	10.19	9.79	9.75
test	*WER*	17.46	7.38	7.53	7.52
	WER-E	13.13	5.86	5.43	5.48
	WER-S	12.53	5.65	5.29	5.25

Table 7.6: Speech Recognition (ASR) performances - ASR Oracle is obtained from 1000-best list by selecting hypothesis that minimizes WER, WER-E or WER-S.

7.4.1 ASR Results

Table 7.6 presents the performances obtained by the ASR system described in section 4.2.2 of chapter 4. The columns correspond to four settings: the best output according to the ASR system, and three oracles extracted from the *N*-best list. The oracle ASR performances are obtained by sorting the *N*-best hypotheses according to WER, WER-E or WER-S. The results show that the oracle hypotheses selected by WER, WER-E and WER-S can be different. In other words, optimizing the ASR according to the new metrics proposed can degrade WER but improve WER-E or WER-S. In this case, better ASR outputs in term of near matches are selected. Overall, whatever the metric used, Oracle hypotheses contain approximately 50% of the initial errors found in the 1-best.

Tasks	*metrics*	Pearson Correlation		
		WER	WER-E	WER-S
dev	*TER*	0.732	0.767	**0.773**
	BLEU	-0.677	-0.708	**-0.710**
	METEOR	-0.753	**-0.799**	-0.797
tst	*TER*	**0.457**	**0.457**	0.441
	BLEU	-0.624	**-0.661**	-0.606
	METEOR	-0.672	**-0.692**	-0.678

Table 7.7: Pearson Correlation between ASR metrics (WER, WER-E or WER-S) and SLT performances (TER, BLEU, METEOR) - each point measured on blocks of 100 sentences.

7.4.2 Correlation between ASR Metrics and SLT Performance

In this section, we investigate if our new metrics WER-E and WER-S are better correlated with speech translation (SLT) performance. Table 7.7 shows the correlation (*Pearson*) between ASR metrics (WER, WER-E or WER-S) and SLT performances (TER, BLEU, METEOR). Since BLEU and METEOR are not very efficient to evaluate translations at the sentence level, we decided to group our sentences by blocks of 100 (in order to have relevant measure points for correlation analysis). We end up with 27 blocks on *dev* and 41 blocks on *test* for evaluating correlation. The reading of the TER score is "the lower the better", and BLEU and METEOR are "the higher the better" which explains the different signs of the correlation values. The results show clearly a better correlation of the proposed metrics (WER-E and WER-S) with SLT performances, compared to classical WER. Also, we notice that all ASR metrics are better correlated with METEOR (itself known to be better correlated with human judgements), while ASR metrics are less correlated with BLEU.

7.4.3 Oracle Analysis

In this section, we verify if the hypotheses selected by WER-E and WER-S are more promising for translation. Our Oracle analysis is presented in Table 7.8. Similarly to Table 7.6, the columns correspond to four settings: the best output according to the ASR system is translated, and three oracles are scored by translating the most promising hypotheses according to WER, WER-E or WER-S. Even if there are not big differences

Tasks	*metrics*	ASR	Oracle from N-best		
		1-best	WER	WER-E	WER-S
dev	*TER*	55.64	50.62	50.52	**50.45**
	BLEU	30.81	35.29	35.37	**35.41**
	METEOR	34.02	36.37	36.42	**36.44**
test	*TER*	58.70	54.13	**54.01**	54.03
	BLEU	34.27	39.34	**39.43**	39.42
	METEOR	34.27	36.55	**36.64**	**36.64**

Table 7.8: Speech Translation (SLT) performances - Oracle is obtained from 1000-best list by translating hypothesis that minimizes WER, WER-E or WER-S.

Tasks	Comparison	TER	BLEU	METEOR
Dev	O. WER-E best	**255**	**310**	**321**
	O. WER best	190	271	315
	Ties	2198	2062	2007
Test	O. WER-E best	**341**	**451**	**510**
	O. WER best	264	381	399
	Ties	3445	3218	3141

Table 7.9: Comparison of SLT performances of the *Oracle WER* vs. the *Oracle WER-E* by counting the number of sentences which obtain a better MT score according to TER, Sentence BLEU and METEOR.

in SLT performance, the results show the ability of our metric to find slightly better hypotheses (to be translated) in the ASR N-best. For instance, when the WER-S score is used to select the best ASR hypothesis, the TER, BLEU and METEOR are improved by respectively 0.18, 0.12, and 0.06 points on the *dev* corpus. However, these differences are rather small.

We also analyzed how often the *Oracle (according to WER-E)* system obtains better results at the sentence level compared to the *Oracle (according to WER)*. Table 7.9 shows this comparison for the three MT metrics (TER, sentenceBLEU and METEOR). Even if we logically observe a majority of ties where *Oracle (according to WER-E)* and *Oracle (according to WER)* lead to the same SLT output, for the other cases the analysis shows a preference of the translation metrics for the *Oracle (according to WER-E)*. This result confirms the trend observed in table 7.8.

Tasks	*metrics*	ASR optimized with WER	ASR optimized with WER-E
dev	*TER*	55.64	**55.52**
	BLEU	30.81	**30.84**
	METEOR	**34.02**	34.00
test	*TER*	58.71	**58.56**
	BLEU	34.27	**34.38**
	METEOR	**34.27**	34.26

Table 7.10: Speech Translation (SLT) scores obtained with 2 ASR systems optimized with WER or WER-E.

7.4.4 ASR Optimization for SLT

This section investigates if the tuning of an ASR system using the new metrics proposed can lead to real (and not oracle) improvements. This experiment is preliminary since we only optimize the LM weight parameter (to minimize WER or WER-E[2]) on the *dev* corpus.

The results are given in table 7.10 but they are not very convincing: we observe small gains for TER and BLEU evaluation but not improvement of METEOR. Our explanation is that there were too few free parameters investigated to tune the ASR system. In addition, translation evaluation metrics are themselves unperfect to evaluate translation quality. The next section proposes to analyze a few translation examples to better understand the differences of both SLT systems.

7.4.5 Translation Examples

In table 7.4 are presented some translation examples related to the ASR optimization. We can observe in these example that both ASR systems (*OptWER* and *OptWER-E*) are very close. For instance, in the first example, the ASR hypothesis is different only on one word (“sera” vs. “serait”). Both are the same verb at the right agreement with the pronoun but not at the same tense. These are two examples where the ASR optimized according to WER-E lead to better translation (SLT) hypotheses than WER. What it means is simply the fact that ASR system is optimized according to a metric which

[2]WER-S lead to the same optimized ASR system than WER-E

penalizes less substitutions between "morphologically similar" words. We believe that for optimizing ASR systems along a larger number of meta-parameters, the modififed metrics proposed in this chapter could be more useful.

7.5 Conclusion

In brief, we proposed an extension of WER in order to penalize differently substitution errors regarding the context.

Our experiments, made on a French-English speech translation task, have shown that the new proposed metric is better correlated with SLT performances. Oracle experiments have also shown a trend: the ability of our metric to find better hypotheses (to be translated) in the ASR N-best. This opens possibilities to optimize ASR using metrics clever than WER. For reproducible experiments, code allowing to call our modified WER has been made available on *github* in collaboration with C.Servan (Post-doc at LIG during this work)[3].

[3] `https://github.com/cservan/tercpp-embeddings`

Chapter 8

Conclusions and Perspectives

8.1 Conclusions

The objective of this thesis was mainly to study a new quality assessment task: word confidence estimation (WCE) for spoken language translation (SLT) that is a sub-field of Confidence Estimation. We proposed several strategies based on several types of features: Machine Translation (MT) based features, Automatic Speech Recognition (ASR) based features, as well as combined or joint features using ASR and MT information.

In addition to the provision of some directions for future research, this thesis has made several contributions to the literature on WCE system for SLT.

First, we extended a speech corpus for a French-English speech translation task. This corpus, which was distributed to the research community[1], now contains 6693 speech recordings (its extension from 2643 to 6693 speech utterances). Duration is 16 hours 52 minutes and it has 42 speakers.

Second, we inherited the conventional ASR and MT features for WCE. We then extracted the full-feature set including new features. We also formalized WCE for SLT, proposed a pipeline of WCE system and developed a complete out-of-the-box toolkit: *LIG-WCE Toolkit* used in this thesis.

[1] `https://github.com/besacier/WCE-SLT-LIG`

Third, we proposed two novel models, which are *combined* model and *joint* model based on SLT features. Those results showed that *joint* model slightly outperforms a model based on MT features only when employing an optimal decision threshold.

Fourth, the results of experiments using *joint* model carried out in this work suggest that there are some redundant predictor features in the full-featured set. This motivate us to employ the "Sequential Backward Selection" (SBS) approach on WCE system for SLT applying *joint* model. When considering the result of feature selection, we could conclude that the most useful are MT features while interesting complementary information can be brought by ASR features.

Fifth, we also experimented with two ASR systems having different performances. The results suggested that WCE performance decreases as ASR system improves.

Sixth, to find out the source of SLT errors, we proposed two methods to disentangle ASR and MT errors in spoken language translation. This was addressed by transforming a 2-class problem into a 3-class problem when labelling our SLT hypotheses. We observed that the task is difficult. But, we hope that the findings of our study could attract the attention of other researchers (not only detecting errors but also their cause).

Finally, in our investigation of tuning SLT output, we proposed a novel metric, called Word Error Rate with Embeddings (WER-E), that could penalize differently substitution errors according to their context using word embeddings. Our experiments showed that ASR hypotheses (*N*-best) optimized with WER-E could help SLT system generate the better candidates. The outcome material of this thesis (corpus, toolkit) can be definitely used to address such a new problem.

8.2 Perspectives

Firstly, we could extend the speech corpora recorded by french native speakers. This task could be important to train robust joint WCE systems for SLT. In addition, more investigation needs to be done in order to disentangle ASR and MT errors in SLT. It is also worth investigating to exploit more in-depth SLT features based on word-level

such as the grammatical content of the word, the relation of the word to the syntactic structure. There are also important directions of potential research that this thesis does not address such as Confidence Estimation (CE) at sentence-level or phrase-level (that are presented in shared task: Quality Estimation of WMT[2]).

As an extension of our proposed Word Error Rate with Embeddings (WER-E) metric, we could replace or augment the word embeddings with lexico-semantic data such as Wordnet or DBnary.

In addition to re-decode SLT graphs, our quality assessment system can be used in scenarios of interactive spoken language translation for example subtitling for lectures, to improve human translator productivity by giving him/her feedback on automatic transcription and translation quality. Another application would be the adaptation of our WCE system to interactive speech-to-speech translation scenarios where feedback on transcription and translation modules is needed to improve communication. Furthermore, we tend to apply some other techniques such as deep learning [Rikters and Fishel, 2017] [Goodfellow et al., 2016] [Lecun et al., 2015] or other ensemble techniques (bagging, voting) to learn and to use the WCE features.

[2] `http://www.statmt.org/wmt17/quality-estimation-task.html`

Appendix A

Extended Summary in French

Les systèmes de traduction de la parole état de l'art commencent à atteindre des performances leur permettant d'être exploités dans des situations réelles. Cependant, ils sont encore confrontés à certaines limites dès que le domaine d'application change des données d'apprentissage. Les mots peu observés ou hors vocabulaire ainsi que les disfluences peuvent avoir des impacts négatifs sur ces systèmes.

Il peut donc être intéressant de pouvoir estimer automatiquement la qualité des sorties d'un système afin d'en extraire des zones de confiance. Cette thèse s'insère donc dans le cadre de l'estimation de mesures de confiance pour la traduction automatique de la parole. Ces travaux pourront ainsi trouver un cadre d'application dans la traduction assistée par ordinateur ou encore la traduction interactive de la parole.

Que ce soit en reconnaissance automatique de la parole ou traduction automatique, il existe de nombreuses approches visant à estimer des mesures de confiance. Elles peuvent être extraites à différentes granularités : au niveau du document [Scarton and Specia, 2014] [Scarton et al., 2016], de la phrase [Blatz et al., 2004] [Specia et al., 2009] [Shah et al., 2016], de segments de mots [Specia and Giménez, 2010] [Logachva and Specia, 2015] [Blain et al., 2016] ou encore au niveau des mots [Ueffing et al., 2003a] [Ueffing and Ney, 2005] [Ueffing and Ney, 2007] [Bach et al., 2011] [Luong et al., 2013a] [Luong et al., 2013b] [Besacier et al., 2014] [Besacier et al., 2015] [Servan et al., 2015] [Logacheva et al., 2016] [Le et al., 2016b].

En raison des causes évoquées précédemment, les sorties d'un système de traduction automatique de la parole peuvent être de qualité insuffisante pour l'utilisation finale. Il est alors nécessaire d'identifier les zones où le système se trompe. Une première contribution de cette thèse est axée autour d'une boîte à outils "LIG-WCE" permettant d'extraire des mesures de confiance au niveau mot. Cette boîte à outils a été faite de manière à être modulable et personnalisable (l'utilisateur peut rajouter des traits supplémentaires facilement).

Bien que les mesures de confiance aient été explorées pour les systèmes de traduction ou de reconnaissance de la parole, peu de travaux ont abordé les mesures de confiance pour la cascade de ces deux types de systèmes. Dans cette thèse, nous formalisons cette tâche comme l'étiquetage de séquence de mots issus du système de traduction automatique de la parole avec des labels "bon" ou "mauvais". Cet étiquetage se fait à l'aide d'un classifieur basé sur des champs aléatoires conditionnels, ayant pour entrée un ensemble de traits internes et externes au système.

Nous proposons plusieurs approches, dans la première nous séparons les estimations de confiance : nous en calculons pour le système de reconnaissance puis pour le système de traduction. Enfin, nous proposons une approche jointe des mesures issues des deux systèmes.

Afin de réaliser nos expériences, nous proposons un corpus contenant 6700 phrases prononcées par différents locuteurs et pour lesquelles sont associés des quintuplets composés de : l'hypothèse du système de reconnaissance, la transcription manuelle, la traduction du verbatim, la traduction du discours et enfin une post-édition de la traduction.

Il ressort de ces expériences que les traits issus du système de traduction automatique sont les plus utiles, tandis que ceux issus du système de reconnaissance automatique de la parole peuvent parfois apporter des informations complémentaires.

Ensuite, nous nous sommes intéressés à identifier automatiquement la provenance des erreurs (parole ou traduction). Nous avons formalisé cette partie en rajoutant des labels "ASR_erreur" et "MT_erreur". Cela nous permet d'identifier l'origine de l'erreur, afin de la corriger en conséquence.

Une dernière contribution est axée sur la proposition d'une nouvelle métrique. Cette dernière propose d'étendre le WER classique afin d'introduire une notion de sémantique : en effet, certaines erreurs de reconnaissance ont peu d'impact sur la traduction car elles restent proches sémantiquement. Cette métrique est basée sur un plongement des mots, qui permet d'identifier les erreurs ayant peu d'impact sémantique. Nous avons notamment réalisé des expériences qui ont montré une forte corrélation entre notre métrique et la qualité du système de traduction de la parole. Les mesures oracles montrent également qu'en se basant sur notre métrique, il est possible de faire remonter de meilleures hypothèses parmi les N-best.

Finalement, nous proposons d'utiliser cette mesure afin d'optimiser notre système de traduction automatique de la parole. Nos expériences montrent un gain significatif grace à ce nouvel estimateur.

En conclusion, nous avons proposé plusieurs stratégies permettant d'extraire des mesures de confiance pour un système de traduction automatique de la parole. Nous avons montré qu'il était possible d'extraire des estimateurs robustes, permettant d'envisager des scénarios de traduction assistée par l'utilisateur (où ce dernier est guidé par les mesures) ou encore de réestimation de graphes de traduction automatique de la parole.

Appendix B

Publications

1. **Ngoc-Tien Le**, Benjamin Lecouteux, and Laurent Besacier. Disentangling ASR and MT Errors in Speech Translation. In *MT Summit XVI – 2017*, Nagoya, Japan, Sep. 2017.

2. **Ngoc-Tien Le**, Benjamin Lecouteux, and Laurent Besacier (Machine Translation Journal - Accepted). Automatic Quality Estimation for Speech Translation Using Joint ASR and MT Features. *CoRR*, abs/1609.06049, 2017. URL `http://arxiv.org/abs/1609.06049`.

3. **Ngoc-Tien Le**, Benjamin Lecouteux, and Laurent Besacier. Joint ASR and MT Features for Quality Estimation in Spoken Language Translation. In *International Workshop on Spoken Language Translation*, Seattle, United States, December 2016. URL `https://hal.archives-ouvertes.fr/al-01408087`.

4. **Ngoc-Tien Le**, Christophe Servan, Benjamin Lecouteux, and Laurent Besacier. Better Evaluation of ASR in Speech Translation Context Using Word Embeddings. In *Interspeech 2016*, San-Francisco, United States, September 2016. URL `https://hal.archives-ouvertes.fr/hal-01350102`.

5. Christophe Servan, **Ngoc-Tien Le**, Ngoc Quang Luong, Benjamin Lecouteux, and Laurent Besacier. An Open Source Toolkit for Word-level Confidence Estimation in Machine Translation. In *The 12th International Workshop on Spoken Language Translation (IWSLT'15)*, Da Nang, Vietnam, December 2015. URL `https://hal.archives-ouvertes.fr/hal-01244477`.

6. Laurent Besacier, Benjamin Lecouteux, Ngoc-Quang Luong, and **Ngoc-Tien Le**. Spoken language translation graphs re-decoding using automatic quality assessment. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scotsdale, Arizona, United States, December 2015. doi: 10.1109/ASRU.2015.7404804. URL `https://hal.archives-ouvertes.fr/hal-01289158`.

Bibliography

V. Alabau, A. Sanchis, and F. Casacuberta. Using word posterior probabilities in lattice translation. In *IWSLT*, pages 131–136, 2007.

D. Arnold and L. D. Tombe. Basic theory and methodology in EUROTRA. In S. Nirenburg, editor, *Machine Translation: Theoretical and Methodological Issues*. Cambridge University Press, Cambridge, UK, 1987.

A. Asadi, R. Schwartz, and J. Makhoul. Automatic detection of new words in a large vocabulary continuous speech recognition system. *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 1990.

N. Bach, F. Huang, and Y. Al-Onaizan. Goodness: A method for measuring machine translation confidence. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 211–219, 2011. URL `http://www.aclweb.org/anthology/P11-1022`.

D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015*, pages 3104–3112, San Diego, California, USA, 2015.

S. Banerjee and A. Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics*, 2005.

F. Bechet, B. Favre, and M. Rouvier. "speech is silver, but silence is golden": improving speech-to-speech translation performance by slashing users input. In *Proceedings of Interspeech 2015*, Dresden, Germany, September 2015.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999. ISBN 9781886529144. URL `https://books.google.fr/books?id=QeweAQAAIAAJ`.

R. C. Berwick and S. Fong. Principle-based parsing: Natural language processing for the 1990s. In P. H. Winston and S. A. Shellard, editors, *Artificial Intelligence at MIT: Expanding Frontiers*, volume 1, chapter 12, pages 287–325. MIT Press, 1990.

L. Besacier, B. Lecouteux, N. Q. Luong, K. Hour, and M. Hadjsalah. Word confidence estimation for speech translation. In *Proceedings of The International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, USA, December 2014.

L. Besacier, B. Lecouteux, N.-Q. Luong, and N.-T. Le. Spoken language translation graphs re-decoding using automatic quality assessment. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scotsdale, Arizona, United States, Dec. 2015. doi: 10.1109/ASRU.2015.7404804. URL https://hal.archives-ouvertes.fr/hal-01289158.

E. Bicici. Referential translation machines for quality estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 343–351, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-2242.

S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

F. Blain, V. Logacheva, and L. Specia. Phrase level segmentation and labelling of machine translation errors. In *Tenth International Conference on Language Resources and Evaluation*, LREC, pages 2240–2245, Portoroz, Slovenia, 2016. ISBN 978-2-9517408-9-1. URL http://www.lrec-conf.org/proceedings/lrec2016/pdf/1194_Paper.pdf.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. Confidence estimation for machine translation. In *Proceedings of COLING 2004*, pages 315–321, Geneva, April 2004.

C. Boitet. Les architectures linguistiques et computationnelles en traduction automatique sont indépendantes. In *Actes de la 15ème conférence sur le Traitement Automatique des Langues Naturelles*, Avignon, France, June 2008. Association pour le Traitement Automatique des Langues. URL http://www.atala.org/taln_archives/TALN/TALN-2008/taln-2008-long-024.

O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, and M. Turchi. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/W15-3001.

T. Brants. Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/974147.974178. URL http://dx.doi.org/10.3115/974147.974178.

L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, Aug. 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350. URL http://dx.doi.org/10.1023/A:1018054314350.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *COMPUTATIONAL LINGUISTICS*, 16(2):79–85, 1990.

P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972470.972474.

C. Callison-Burch, M. Osborne, and P. Koehn. Re-evaluating the role of bleu in machine translation research. In *Proceedings of EACL*, volume 2006, pages 249–256, 2006.

C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June 2012. Association for Computational Linguistics.

M. Candito, J. Nivre, P. Denis, and E. H. Anguiano. Benchmarking of statistical dependency parsers for french. In *Proceedings of COLING'2010*, 2010.

J. Carbonell, S. Klein, D. Miller, M. Steinbaum, T. Grassiany, and J. Frey. Context-based machine translation. In *In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 8–12, 2006.

J. G. Carbonell, T. Mitamura, and E. H. Nyberg. The kant perspective: A critique of pure transfer- and pure interlingua, pure transfer,... In *Proc. of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation: Empiricist vs. Rationalist Methods in MT*, pages 225–235, Montreal, Canada, 1992.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. doi: 10.3115/981863.981904. URL http://dx.doi.org/10.3115/981863.981904.

M. R. Costa-jussá and J. A. Fonollosa. Latest trends in hybrid machine translation and its applications. *Comput. Speech Lang.*, 32(1):3–10, July 2015. ISSN 0885-2308. doi: 10.1016/j.csl.2014.11.001. URL http://dx.doi.org/10.1016/j.csl.2014.11.001.

G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, Jan 2012. ISSN 1558-7916. doi: 10.1109/TASL.2011.2134090.

S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *ACOUSTICS, SPEECH AND SIGNAL PROCESSING, IEEE TRANSACTIONS ON*, pages 357–366, 1980.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

P. R. Dixon, A. Finch, C. Hori, and H. Kashioka. Investigation on the effects of ASR tuning on speech translation performance. In *The proceedings of the International Workshop on Spoken Language Translation (IWSLT 2011)*, San Francisco, December 2011.

L. Dugast, J. Senellart, and P. Koehn. Can we relearn an rbmt system? In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 175–178, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1. URL `http://dl.acm.org/citation.cfm?id=1626394.1626421`.

J. L. Elman. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211, 1990.

C. España-Bonet and M. R. Costa-jussà. Hybrid machine translation overview. In *Hybrid Approaches to Machine Translation*, pages 1–24. Springer, 2016.

B. Favre, K. Cheung, S. Kazemian, A. Lee, Y. Liu, C. Munteanu, A. Nenkova, D. Ochei, G. Penn, S. Tratz, C. Voss, and F. Zeller. Automatic Human Utility Evaluation of ASR Systems: Does WER Really Predict Performance? In *Proceedings of Interspeech 2013*, Lyon, France, August 2013.

J. Fayolle, F. Moreau, C. Raymond, G. Gravier, and P. Gros. Crf-based combination of contextual features to improve a posteriori word-level confidence measures. In *Interspeech*, 2010.

M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker. Overview of the IWSLT 2012 evaluation campaign. In *In proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, December 2012.

M. Felice and L. Specia. Linguistic features for quality estimation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 96–103, Montreal, Canada, June 7-8 2012.

A. J. Ferreira and M. A. T. Figueiredo. *Boosting Algorithms: A Review of Methods, Theory, and Applications*, pages 35–85. Springer US, Boston, MA, 2012. ISBN 978-1-4419-9326-7. doi: 10.1007/978-1-4419-9326-7_2. URL `http://dx.doi.org/10.1007/978-1-4419-9326-7_2`.

G. D. Forney. The viterbi algorithm. *Proc. of the IEEE*, 61:268 – 278, March 1973.

E. Frank and R. R. Bouckaert. Naive bayes for text classification with unbalanced classes. In *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases*, PKDD'06, pages 503–510, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-45374-1, 978-3-540-45374-1. doi: 10.1007/11871637_49. URL http://dx.doi.org/10.1007/11871637_49.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256 – 285, 1995. ISSN 0890-5401. doi: http://dx.doi.org/10.1006/inco.1995.1136. URL http://www.sciencedirect.com/science/article/pii/S0890540185711364.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996)*, pages 148–156. Morgan Kaufmann, 1996. ISBN 1-55860-419-7. URL http://www.biostat.wisc.edu/~kbroman/teaching/statgen/2004/refs/freund.pdf.

Y. Freund and R. E. Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

S. Galliano, E. Geoffrois, G. Gravier, J.-F. Bonastre, D. Mostefa, and K. Choukri. Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *In Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC 2006)*, pages 315–320, 2006.

Y. Goldberg. A primer on neural network models for natural language processing. *J. Artif. Intell. Res. (JAIR)*, 57:345–420, 2016. doi: 10.1613/jair.4992. URL http://dx.doi.org/10.1613/jair.4992.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3 and 4):237–264, 1953.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

A. Guerra. *Machine Translation. Capabilities and limitations*. SELL monographs. Lengua Inglesa, Univ., 2000. ISBN 9788437042664. URL https://books.google.fr/books?id=7TE3avRZiSoC.

H. A. Güvenir and I. Cicekli. Learning translation templates from examples. *Information systems*, 23(6):353–363, 1998.

A. L.-F. Han, Y. Lu, D. F. Wong, L. S. Chao, L. He, and J. Xing. Quality estimation for machine translation using the joint method of evaluation criteria and statistical modeling. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 365–372, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-2245.

X. He, L. Deng, and A. Acero. Why word error rate is not a good metric for speech recognizer training for the speech translation task? In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5632–5635, May 2011. doi: 10.1109/ICASSP.2011.5947637.

H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *J. Acoust. Soc. Am.*, 57(4):1738–52, Apr. 1990.

G. Hinton. A Practical Guide to Training Restricted Boltzmann Machines. Technical report, 2010. URL https://www.researchgate.net/publication/272746011_A_Practical_Guide_to_Training_Restricted_Boltzmann_Machines_Version_1.

G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527. URL http://dx.doi.org/10.1162/neco.2006.18.7.1527.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL http://arxiv.org/abs/1207.0580.

T. K. Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7128-9. URL http://dl.acm.org/citation.cfm?id=844379.844681.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, Aug. 1998. ISSN 0162-8828. doi: 10.1109/34.709601. URL http://dx.doi.org/10.1109/34.709601.

S. Hochreiter, Y. Bengio, and P. Frasconi. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In J. Kolen and S. Kremer, editors, *Field Guide to Dynamical Recurrent Networks*. IEEE Press, 2001.

W. J. Hutchins. *Machine Translation: Past, Present, Future*. John Wiley & Sons, Inc., New York, NY, USA, 1986. ISBN 0-470-20313-7.

W. J. Hutchins. Machine translation: A brief history. In *Concise history of the language sciences: from the Sumerians to the cognitivists, Pergamon*, pages 431–445. Press, 1995.

W. J. Hutchins. Machine translation: a concise history. In *Computer aided translation: theory and practice*, 2007.

S. Jalalvand, M. Negri, M. Turchi, J. G. C. de Souza, F. Daniele, and M. R. H. Qwaider. Transcrater: a tool for automatic speech recognition quality estimation. In *Proceedings of ACL-2016 System Demonstrations*, pages 43–48, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/P16-4008.

F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings Workshop on Pattern Recognition in Practice, 1980*, pages 381–397, 1980.

D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000. ISBN 0130950696.

N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. Seattle, October 2013. Association for Computational Linguistics.

S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 400–401, 1987.

M. Kearns and L. G. Valiant. Crytographic limitations on learning boolean formulae and finite automata. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 433–444, New York, NY, USA, 1989. ACM. ISBN 0-89791-307-8. doi: 10.1145/73007.73049. URL `http://doi.acm.org/10.1145/73007.73049`.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 181–184, May 1995.

P. Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073462. URL `http://dx.doi.org/10.3115/1073445.1073462`.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic, June 2007.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, Californie, États-Unis d'Amérique, 2001.

D. Langlois, S. Raybaud, and K. Smaïli. Loria system for the wmt12 quality estimation shared task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 114–119, Baltimore, Maryland USA, June 2012.

A. Laurent, N. Camelin, and C. Raymond. Boosting bonsai trees for efficient features combination : application to speaker role identification. In *Interspeech*, 2014a.

A. Laurent, N. Camelin, and C. Raymond. Boosting bonsai trees for efficient features combination: application to speaker role identification. In *InterSpeech*, pages 76–80, Singapour, September 2014b.

T. Lavergne, O. Cappé, and F. Yvon. Practical very large scale crfs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513, 2010.

N. Le, B. Lecouteux, and L. Besacier. Automatic quality assessment for speech translation using joint ASR and MT features. *CoRR*, abs/1609.06049, 2016a. URL `http://arxiv.org/abs/1609.06049`.

N.-T. Le, B. Lecouteux, and L. Besacier. Joint ASR and MT Features for Quality Estimation in Spoken Language Translation. In *International Workshop on Spoken Language Translation*, Seattle, United States, Dec. 2016b. URL `https://hal.archives-ouvertes.fr/hal-01408087`.

N.-T. Le, C. Servan, B. Lecouteux, and L. Besacier. Better Evaluation of ASR in Speech Translation Context Using Word Embeddings. In *Interspeech 2016*, San-Francisco, United States, Sept. 2016c. URL `https://hal.archives-ouvertes.fr/hal-01350102`.

N.-T. Le, B. Lecouteux, and L. Besacier. Disentangling ASR and MT Errors in Speech Translation. In *MT Summit 2017*, Nagoya, Japan, Sept. 2017. URL `https://hal.archives-ouvertes.fr/hal-01580877`.

B. Lecouteux, G. Linarès, and B. Favre. Combined low level and high level features for out-of-vocabulary word detection. *INTERSPEECH*, 2009.

Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 5 2015. ISSN 0028-0836. doi: 10.1038/nature14539.

S. Levinson, L. Rabiner, and M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell Labs Technical Journal*, 62(4):1035–1074, 1983. ISSN 1089-7089. doi: 10.1002/j.1538-7305.1983.tb03114.x.

Z. H. Ling, S. Y. Kang, H. Zen, A. Senior, M. Schuster, X. J. Qian, H. M. Meng, and L. Deng. Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine*, 32(3):35–52, May 2015. ISSN 1053-5888. doi: 10.1109/MSP.2014.2359987.

V. Logacheva, M. Lukasik, and L. Specia. Metrics for evaluation of word-level machine translation quality estimation. In *54th Annual Meeting of the Association for Computational Linguistics*, ACL, Berlin, Germany, 2016.

V. Logachva and L. Specia. Phrase-level quality estimation for machine translation. In *Conference on Empirical Methods in Natural Language Processing*, IWSLT, pages 143–150, Da Nang, Vietnam, 2015. URL `http://workshop2015.iwslt.org/downloads/IWSLT_2015_RP_12.pdf`.

L. Lu, A. Ghoshal, and S. Renals. Regularized subspace gaussian mixture models for speech recognition. *IEEE signal processing letters*, 18(7):419–422, 2011. DK.

N. Q. Luong, L. Besacier, and B. Lecouteux. Word confidence estimation and its integration in sentence quality estimation for machine translation. In *Proceedings of The Fifth International Conference on Knowledge and Systems Engineering (KSE 2013)*, Hanoi, Vietnam, October 17-19 2013a.

N. Q. Luong, B. Lecouteux, and L. Besacier. LIG system for WMT13 QE task: Investigating the usefulness of features in word confidence estimation for MT. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 396–391, Sofia, Bulgaria, August 2013b. Association for Computational Linguistics.

N.-Q. Luong, L. Besacier, and B. Lecouteux. Word Confidence Estimation for SMT N-best List Re-ranking. In *Proceedings of the Workshop on Humans and Computer-assisted Translation (HaCaT) during EACL*, Gothenburg, Suède, 2014a. URL `http://hal.inria.fr/hal-00953719`.

N.-Q. Luong, L. Besacier, and B. Lecouteux. LIG System for Word Level QE task at WMT14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 335–341, Baltimore, Maryland USA, June 2014b.

N.-Q. Luong, L. Besacier, and B. Lecouteux. Towards accurate predictors of word quality for machine translation: Lessons learned on french - english and english - spanish systems. *Data and Knowledge Engineering*, page 11, Apr. 2015.

T. Luong, K. Cho, and C. D. Manning. Neural machine translation. Tutorials - ACL 2016, Berlin, Germany, 2016. URL `http://acl2016.org/index.php?article_id=55`.

B. MacCartney. Nlp lunch tutorial: Smoothing, 2005. URL `http://nlp.stanford.edu/~wcmac/papers/20050421-smoothing-tutorial.pdf`.

Y. Mansour. Pessimistic decision tree pruning based on tree size. In *Proc. 14th International Conference on Machine Learning*, pages 195–201. Morgan Kaufmann, 1997.

J. Mauclair, Y. Estève, S. Petit-renaud, and P. Deléglise. Automatic detection of well recognized words in automatic speech transcription. In *LREC 2006*, Genoa (Italy), 24-26 may 2006.

A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press, 1998.

V. Metsis and et al. Spam filtering with naive bayes – which naive bayes? In *THIRD CONFERENCE ON EMAIL AND ANTI-SPAM (CEAS*, 2006.

T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010. URL http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *The Workshop Proceedings of the International Conference on Learning Representations (ICLR) 2013*, Scottsdale, Arizona, USA, May 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013b.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013c.

G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

K. Miyahara and M. J. Pazzani. Collaborative filtering with the simple bayesian classifier. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, PRICAI'00, pages 679–689, Berlin, Heidelberg, 2000. Springer-Verlag. ISBN 3-540-67925-1. URL http://dl.acm.org/citation.cfm?id=1764967.1765055.

M. Nagao. A framework of a mechanical translation between japanese and english by analogy principle. *Artificial and human intelligence*, pages 351–354, 1984.

M. Nagao. *Machine translation: how far can it go?* Oxford University Press, 1989. ISBN 9780198537397. URL https://books.google.co.in/books?id=WkdiAAAAMAAJ.

R. Navigli and S. P. Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86,

Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118704. URL https://doi.org/10.3115/1118693.1118704.

K. Papineni, S. Roukos, T. Ard, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

S. Petrov and D. Klein. Improved inference for unlexicalized parsing. In *HLT-NAACL*, 2007.

M. Potet, L. Besacier, and H. Blanchon. The lig machine translation system for wmt 2010. In A. Workshop, editor, *Proceedings of the joint fifth Workshop on Statistical Machine Translation and Metrics MATR (WMT2010)*, Uppsala, Sweden, 11-17 July 2010.

M. Potet, R. Emmanuelle E, L. Besacier, and H. Blanchon. Collection of a large database of french-english smt output corrections. In *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, May 2012.

D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas. The subspace gaussian mixture model-a structured model for speech recognition. *Comput. Speech Lang.*, 25(2):404–439, Apr. 2011a. ISSN 0885-2308. doi: 10.1016/j.csl.2010.06.003. URL http://dx.doi.org/10.1016/j.csl.2010.06.003.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011b. IEEE Catalog No.: CFP11SRW-USB.

J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, Mar. 1986. ISSN 0885-6125. doi: 10.1023/A:1022643204877. URL http://dx.doi.org/10.1023/A:1022643204877.

J. R. Quinlan. Simplifying decision trees. *Int. J. Man-Mach. Stud.*, 27(3):221–234, Sept. 1987. ISSN 0020-7373. doi: 10.1016/S0020-7373(87)80053-6. URL http://dx.doi.org/10.1016/S0020-7373(87)80053-6.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-015157-2.

L. R. Rabiner. Readings in speech recognition. In A. Waibel and K.-F. Lee, editors, *PROCEEDINGS OF THE IEEE*, chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-124-4. URL http://dl.acm.org/citation.cfm?id=108235.108253.

S. Rao, I. Lane, and T. Schultz. Improving spoken language translation by automatic disfluency removal: Evidence from conversational speech transcripts. In *In Machine Translation Summit XI*, 2007.

S. Raybaud, D. Langlois, and K. Smaili. "this sentence is wrong." detecting errors in machine-translated sentences. *Machine Translation*, 25(1):p. 1–34, Aug. 2011. doi: 10.1007/s10590-011-9094-9.

C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

M. Rikters and M. Fishel. Confidence through Attention. In *MT Summit 2017*, Nagoya, Japan, Sept. 2017. URL `http://arxiv.org/abs/1710.03743`.

I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

F. Rosenblatt. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books, 1962. URL `https://books.google.fr/books?id=7FhRAAAAMAAJ`.

R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here. In *Proceedings of the IEEE*, page 2000, 2000.

N. Ruiz and M. Federico. Phonetically-oriented word error alignment for speech recognition error analysis in speech translation. In *IEEE 2015 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2015.

C. Scarton and L. Specia. Document-level translation quality estimation: exploring discourse and pseudo-references. In *EAMT 2014*, pages 101–108, Dubrovnik, Croatia, 2014.

C. Scarton, D. Beck, K. Shah, K. Sim Smith, and L. Specia. Word embeddings and discourse information for quality estimation. In *Proceedings of the First Conference on Machine Translation*, pages 831–837, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W16/W16-2391`.

S. Schachter, N. Christenfeld, B. Ravina, and F. Bilous. Speech disfluency and the structure of knowledge. *Journal of Personality and Social Psychology*, 60(3):362, 1991.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990. ISSN 1573-0565. doi: 10.1007/BF00116037. URL `http://dx.doi.org/10.1007/BF00116037`.

H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, 1994.

H. Schmid. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, Dublin, Ireland, 1995.

N. Segal, H. Bonneau-Maynard, and F. Yvon. Traduire la parole : le cas des TED talks. *TAL (Traitement Automatique des Langues)*, 55(2):13–45, 2015. URL http://atala.org/IMG/pdf/1.Maynard-TAL55-2.pdf.

F. Seide, G. Li, and D. Yu. Conversational speech transcription using context-dependent deep neural networks. In *in Proc. Interspeech '11*, pages 437–440, 2011.

G. Sérasset. DBnary: Wiktionary as a Lemon-Based Multilingual Lexical Resource in RDF. *Semantic Web – Interoperability, Usability, Applicability*, pages –, 2014. URL https://hal.archives-ouvertes.fr/hal-00953638. To appear.

C. Servan, N.-T. Le, N. Q. Luong, B. Lecouteux, and L. Besacier. An Open Source Toolkit for Word-level Confidence Estimation in Machine Translation. In *The 12th International Workshop on Spoken Language Translation (IWSLT'15)*, Da Nang, Vietnam, Dec. 2015. URL https://hal.archives-ouvertes.fr/hal-01244477.

K. Shah, F. Bougares, L. Barrault, and L. Specia. Shef-lium-nn: Sentence level quality estimation with neural network features. In *First Conference on Machine Translation, Volume 2: Shared Task Papers*, WMT, pages 828–832, Berlin, Germany, 2016. URL http://www.aclweb.org/anthology/W/W16/W16-2391.

J. Slocum. A survey of machine translation: Its history, current status, and future prospects. *Comput. Linguist.*, 11(1):1–17, Jan. 1985. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=5615.5616.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231, 2006.

M. Snover, N. Madnani, B. Dorr, and R. Schwartz. Terp system description. In *MetricsMATR workshop at AMTA*, 2008.

H. Somers. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157, June 1999. ISSN 0922-6567. doi: 10.1023/A:1008109312730. URL http://dx.doi.org/10.1023/A:1008109312730.

L. Specia and J. Giménez. Combining confidence estimation and reference-based metrics for segment-level mt evaluation. In *Ninth Conference of the Association for Machine Translation in the Americas*, AMTA, Denver, Colorado, 2010. URL http://www.mt-archive.info/AMTA-2010-Specia.pdf.

L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 18–37, Barcelona, 2009.

F. Stella and Y. Amer. Continuous time bayesian network classifiers. *Journal of Biomedical Informatics*, 45(6):1108 – 1119, 2012. ISSN 1532-0464. doi: http://doi.org/10.1016/j.jbi.2012.07.002. URL http://www.sciencedirect.com/science/article/pii/S1532046412000998.

A. Stolcke. Srilm - an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, USA, 2002.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969173.

C. Sutton and A. McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, Apr. 2012. ISSN 1935-8237. doi: 10.1561/2200000013. URL http://dx.doi.org/10.1561/2200000013.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 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*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073478. URL http://dx.doi.org/10.3115/1073445.1073478.

J. E. Tree. The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech. *Journal of Memory and Language*, 34(6):709 – 738, 1995. ISSN 0749-596X. doi: http://dx.doi.org/10.1006/jmla.1995.1032. URL http://www.sciencedirect.com/science/article/pii/S0749596X85710327.

C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro. Sentiment analysis of facebook statuses using naive bayes classifier for language learning. In *IISA 2013*, pages 1–6, July 2013. doi: 10.1109/IISA.2013.6623713.

N. Ueffing and H. Ney. Word-level confidence estimation for machine translation using phrased-based translation models. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 763–770, Vancouver, 2005.

N. Ueffing and H. Ney. Word-level confidence estimation for machine translation. In *Computational Linguistics*, volume 33, pages 9–40, 2007.

N. Ueffing, K. Macherey, and H. Ney. Confidence measures for statistical machine translation. In *Proceedings of the MT Summit IX*, pages 394–401, New Orleans, LA, September 2003a.

N. Ueffing, K. Macherey, and H. Ney. Confidence measures for statistical machine translation. In *Proceedings of the MT Summit IX*, pages 394–401, New Orleans, LA, September 2003b.

B. Vauquois. A survey of formal grammars and algorithms for recognition and transformation in machine translation. In *IFIP Congress-68*, pages 254–260, Edinburgh, 1968.

D. Vilar, J. Xu, L. D'haro, and H. Ney. Error analysis of statistical machine translation output. In *Proceedings of the Fifth International Conference on Language Resources*

and Evaluation (LREC-2006), Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/413_pdf.pdf. ACL Anthology Identifier: L06-1244.

A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theor.*, 13(2):260–269, Sept. 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1054010. URL http://dx.doi.org/10.1109/TIT.1967.1054010.

K. Wang and Y. Tan. *A New Collaborative Filtering Recommendation Approach Based on Naive Bayesian Method*, pages 218–227. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-21524-7. doi: 10.1007/978-3-642-21524-7_26. URL http://dx.doi.org/10.1007/978-3-642-21524-7_26.

W. Weaver. Translation. In W. N. Locke and A. D. Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA, 1949. Reprinted from a memorandum written by Weaver in 1949.

F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, 2001.

C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30:79–82, 2005. URL http://www.int-res.com/abstracts/cr/v30/n1/p79-82/.

I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Inf. Theor.*, 37(4):1085–1094, 1991. ISSN 0018-9448. doi: 10.1109/18.87000. URL http://dx.doi.org/10.1109/18.87000.

D. Xiong, M. Zhang, and H. Li. Error detection for statistical machine translation using linguistic features. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, pages 604–611, Uppsala, Sweden, July 2010.

S. R. Young. Recognition confidence measures: Detection of misrecognitions and out-of-vocabulary words. *Proc. of International Conference on Acoustics, Speech and Signal Processing*, pages 21–24, 1994.