Word Sense Consistency in Statistical and Neural Machine Translation

THÈSE N^{O} 8743 (2018)

PRÉSENTÉE LE 31 AOÛT 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE L'IDIAP
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Xiao PU

acceptée sur proposition du jury:

Prof. P. Frossard, président du jury
Prof. H. Bourlard, Prof. A. Popescu-Belis, directeurs de thèse
Prof. P. Merlo, rapporteuse
Prof. M. Volk, rapporteur
Dr M. Rajman, rapporteur

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2018

Acknowledgments

Look deep into nature,
and then you will understand everything better.
Albert Einstein

I would like to express my deepest gratitude to my advisors Andrei Popescu-Belis and Hervé Bourlard for their kind inspiration, consultation and understanding. Andrei supported my work in every thinkable way and I could not have been able to complete this work without his patience, motivation and constant input.

I am also grateful to the Swiss National Science Foundation (SNSF) for its financial support through the MODERN Sinergia project on "Modeling Discourse Entities and Relations for Coherent Machine Translation." Furthermore, the Idiap Research Institute and all its staff provided an environment of friendliness, flexibility and enjoyable learning.

I am thankful to my thesis committee, Professors Pascal Frossard, Paola Merlo, Martin Rajman, and Martin Volk, who reviewed the present thesis in great detail and provided insightful remarks and comments that greatly improved the quality and clarity of the final version.

At Idiap, I would like to thank my colleagues from the NLP group: Lesly, Nikos, and Jamie, now leader of the NLU group. They provided personal, psychological and scientific support throughout the years. The moments of leisure shared together helped to overcome some more difficult moments.

To my closer friends, I express my special gratitude for their unconditional friendship, support, patience throughout these years.

Last, but not least, I would like to thank my family, my parents and my husband, for their unconditional support, encouragement and love, without which I would not have come this far.

Lausanne, July 2018 X. P.

Abstract

Different senses of source words must often be rendered by different words in the target language when performing machine translation (MT). Selecting the correct translation of polysemous words can be done based on the contexts of use. However, state-of-the-art MT algorithms generally work on a sentence-by-sentence basis that ignores information across sentences. In this thesis, we address this problem by studying novel contextual approaches to reduce source word ambiguity in order to improve translation quality. The thesis consists of two parts: the first part is devoted to methods for correcting ambiguous word translations by enforcing consistency across sentences, and the second part investigates sense-aware MT systems that address the ambiguity problem for each word.

In the first part, we propose to reduce word ambiguity by using lexical consistency, starting from the one-sense-per-discourse hypothesis. If a polysemous word appears multiple times in a discourse, it is likely that its occurrences will share the same sense. We first improve the translation of polysemous nouns in the case when an occurrence of a noun as the head of a compound noun phrase is available in a text. We propose to identify through pattern matching rules the occurrences of XY compounds followed closely by a potentially co-referent occurrence of Y. We compare two strategies (cache-based vs. post-editing) to improve the translation of the second occurrence of Y. Experiments on two language pairs show that the translations of the targeted polysemous nouns are significantly improved.

To go beyond the condition of compound pairs XY/Y, we extend our work by analyzing the repetition of nouns which are not compounds. We propose a method to decide whether two occurrences of the same noun in a source text should be translated consistently. We design a classifier to predict translation consistency based on lexical, syntactic, and semantic features. We integrate the classifiers' output into MT by post-editing and/or re-ranking the translation hypothesis. We experiment on two language pairs and show that our method closes up to 50% of the gap in BLEU scores between the baseline and an oracle classifier.

In the second part of the thesis, we design sense-aware MT systems that select the correct translations of ambiguous words by performing word sense disambiguation (WSD). We demonstrate that WSD improves MT by widening the source context that is considered when modeling the senses of potentially ambiguous words. We propose three adaptive clustering algorithms, respectively based on k-means, the Chinese restaurant process and random walks. For phrase-based statistical MT, we integrate the sense knowledge as an additional feature through a factored model and show that this improves noun and verb translation from English to five other languages.

As the sense integration appears promising for SMT, we also transfer this approach to the newer neural MT models (NMT), which are now state of the art. However, unlike SMT, for which it is easier to use linguistic features, NMT uses vectors for word generation and traditional feature incorporation does not work here. We design a sense-aware NMT model that jointly learns the sense knowledge using an attention-based sense selection mechanism and concatenates the learned sense vectors with word vectors during encoding . Such a concatenation outperforms several baselines, including other sense-aware NMT systems. The improvements are significant for all of the language pairs used in the SMT experiments.
Overall, the thesis proves that lexical consistency and WSD are practical and workable solutions that lead to global improvements in translation in ranges of 0.2 to 1.5 BLEU score.

Keywords: Statistical Machine Translation, Neural Machine Translation, Word Sense Disambiguation, Lexical Consistency, Discourse

Résumé

Lorsqu'un mot possède plusieurs sens différents, les traductions de ces sens peuvent utiliser des mots différents. Les systèmes de traduction automatique actuels travaillent en général phrase par phrase, et ne propagent pas l'information d'une phrase à une autre, alors que cette propagation du contexte permettrait d'améliorer les choix de traduction des mots polysémiques. Dans cette thèse, afin de proposer des solutions à ce problème, nous étudions des approches novatrices qui utilisent le contexte pour réduire l'ambiguïté des mots dans les textes source, et améliorer ainsi la qualité des traductions automatiques. La thèse est organisée en deux parties : la première est consacrée aux méthodes qui corrigent les traductions des mots ambigus en exploitant la cohérence entre propositions, et la seconde étudie des systèmes de traduction automatique prenant en compte les sens des mots pour réduire l'ambiguïté de chaque occurrence.

Dans la première partie, nous proposons d'utiliser la cohérence lexicale pour traiter les mots ambigus, faisant l'hypothèse que si un mot polysémique apparaît plusieurs fois dans un document, alors il est probable que ses occurrences auront toutes le même sens. Nous améliorons d'abord la traduction des noms polysémiques lorsque ceux-ci apparaissent d'abord au sein d'un mot composé, puis en isolation. Nous proposons des règles pour identifier les occurrences de mots composés XY suivis d'une occurrence de Y (la tête lexicale) potentiellement coréférente. Nous comparons deux stratégies pour améliorer la traduction de la seconde occurrence de Y, à savoir l'utilisation d'un cache et la post-édition. Les expériences avec deux paires de langues montrent que les traductions sont effectivement améliorées.

Nous étendons ensuite notre proposition aux noms qui se répètent sans faire partie de mots composés, ce qui nous permet de traiter des configurations plus fréquentes que les précédentes. Nous proposons pour ce faire une méthode pour décider si deux occurrences d'un même nom dans un même texte doivent ou non être traduites de la même façon, en utilisant un classifieur avec des traits lexicaux, syntaxiques ou sémantiques. Les prédictions de ce classifieur sont utilisées en traduction automatique soit via la post-édition, soit en réordonnant les traductions candidates. Nos expériences avec deux paires de langues montrent que notre meilleure méthode arrive à la moitié de la distance selon le score BLEU entre un système de base et un oracle.

Dans la seconde partie de la thèse, nous proposons des systèmes de traduction automatique qui choisissent la traduction correcte de mots polysémiques grâce à la désambiguïsation sémantique. Nous montrons que celle-ci peut améliorer la traduction en élargissant le contexte source pris en compte pour modéliser le sens des mots potentiellement ambigus. Nous présentons trois algorithmes de *clustering* des occurrences des mots, basés respectivement sur les méthodes des k-moyennes, le processus du restaurant chinois, et les marches aléatoires. Pour un système de traduction automatique statistique à base de n-grammes, nous ajoutons l'étiquette de sens comme un trait supplémentaire dans un modèle factorisé. Cela améliore la traduction des noms et des verbes de l'anglais vers cinq autres langues.

Dans la mesure où cette approche est prometteuse, nous la transférons également vers des systèmes de traduction neuronale, qui sont devenus l'état de l'art. Au contraire des systèmes à base de n-grammes, dans lesquels l'ajout de traits linguistiques est relativement aisé, les systèmes neuronaux utilisent des vecteurs de mots, et il n'est pas possible d'ajouter des traits de manière traditionnelle. Nous proposons alors un modèle neuronal qui peut apprendre des informations sémantiques en utilisant l'attention du réseau, et concatène les vecteurs de sens avec les vecteurs des mots obtenus dans l'encodeur du réseau. Ce modèle dépasse les modèles de base ainsi qu'un autre modèle représentant le sens lexical. Les améliorations sont significatives, que ce soit en moyenne sur tous les mots, ou sur les mots ambigus, sur les mêmes paires de langues que ci-dessus.

Dans son ensemble, la thèse démontre que la cohérence lexicale et la désambiguïsation sémantique sont utilisables de manière réaliste pour la traduction automatique et permettent d'obtenir des améliorations de la traduction de l'ordre de 0.2 à 1.5 points de score BLEU.

Mots clés : traduction automatique statistique, traduction automatique neuronale, désambiguïsation lexicale, cohérence lexicale, discours

Contents

List of Figures

List of Tables

1 Introduction

Machine translation (MT) is the translation of texts by a computer, with no human involvement. The first MT systems were rule-based ones, using a combination of language and grammar rules plus dictionaries for common words. Specialist dictionaries could be added to focus on certain industries or domains. Rule-based systems could typically deliver consistent translations with accurate terminology when used with specialist dictionaries.

Later on, the field moved to statistical machine translation (SMT) systems, which do not have initial knowledge of any language rules. SMT systems can also be trained for specific industries or domains by using additional data relevant to these domains. Typically, SMT systems deliver more fluent-sounding but less consistent translations than rule-based ones.

In recent years, neural networks became popular and successful in many research areas, including MT. Neural Machine Translation (NMT) systems apply machine learning to language translation through one large neural network. This approach has become increasingly popular among MT researchers and developers, as trained NMT systems have begun to show better translation performance in many language pairs compared to the SMT approach.

All MT systems, from rule-based to recent neural MT ones, translate sentence by sentence and consider sentences in isolation. However, words tend to appear as more ambiguous if the context is reduced to the current sentence. The aim of this thesis is to reduce the word ambiguity by considering a larger context in statistical and neural MT.

The thesis consists of two parts: the first part is devoted to methods for correcting ambiguous word translations by using lexical consistency across sentences, and the second part investigates sense-aware MT systems that address word ambiguity thanks to contextual adaptive word sense disambiguation.

1.1 Summary of Contributions

This thesis makes the following two major contributions to improve translation at the word level, by using lexical consistency and word sense disambiguation.

- First we design several approaches to check and correct the translation of ambiguous words from a baseline translation system, based on the one-sense-per-discourse hypothesis.
  - We correct the translation of uncertain words by enforcing the consistency of compounds.
  - Then, we extend the solution to the translation of repeated nouns, deciding whether to enforce consistency through a machine learning approach with syntactic and semantic features.
- As word post-editing appeared promising for MT, in the second part of the thesis, we design sense-aware SMT and NMT systems which do not require the repetition of a word to attempt to improve its translation.
  - First, we address word sense ambiguity on the source side during decoding, using factored models in statistical phrase-based MT.
  - Finally, we design a sense-aware NMT model that jointly learns the senses of ambiguous words and selects the correct translation during decoding.

Therefore, the thesis consists of two major parts, presenting the above contributions respectively. The thesis is organized as follows. We first provide a detailed overview of SMT and NMT models and the related knowledge that is used in the thesis in Chapter 2. Then, the two parts form the main body of the thesis, each starting with a section describing related studies and ending with a section of conclusions. Finally, we conclude our studies and present perspectives in Chapter 10. We outline below the contributions of each of the parts.

1.1.1 Lexical Consistency for MT

In the first part of our thesis, we start by applying the disambiguation potential of compound words to subsequent occurrences of their individual components. As shown in Chapter 4, we assume that the translation of a noun-noun compound, noted XY, displays fewer ambiguities than the separate translations of its components X and Y. Therefore, on a subsequent occurrence of the head Y of XY, assumed to refer to the same entity as XY, we hypothesize that its previously-found translation offers a better and more coherent translation than the one proposed by an SMT system that is not aware of the compound.

The main components of this system are the rules for identifying XY/Y pairs, and two alternative methods for improving the coherence of the translation of a subsequent mention Y

– one based on post-editing and the other one based on caching, which builds upon initial experiments presented by Mascarell et al. [2014] within the MODERN project (see Section 1.2 below). We evaluate our proposal on Chinese-to-English and German-to-French translation, demonstrating that the translation of nouns is indeed improved, as shown by automatic or human comparisons with the reference translation.

As compound pairs XY/Y appear quite infrequently in texts, we extend our work by considering the repetition of nouns – a more general case than the repetition of compound heads. The repetition of a noun in a text may be due to co-reference, i.e. repeated mentions of the same entity. But in other cases, two occurrences of the same noun may simply convey different meanings. The translation of repeated nouns depends, among other things, on these meanings: in case of co-reference or identical senses, the nouns should likely be translated with the same word, while otherwise they should be translated with different words, if the target language distinguishes the two meanings. State-of-the-art MT systems do not address this challenge systematically, and translate two occurrences of the same noun independently, thus potentially introducing unwanted variations in translation.

In Chapter 5, we aim to improve the translation of repeated nouns by designing a classifier which predicts, for every pair of repeated nouns in a source text, whether they should be translated by the same noun, i.e. consistently, and if that is the case, which of the two candidate translations generated by an MT system should replace the other one. We thus address one kind of long-range dependencies between words in texts, among others that have been recently studied (which we review in Chapter 3).

To learn a consistency classifier from the data, we consider a corpus with source texts in German and Chinese and reference translations into English. We mine the corpus for pairs of repeated nouns in the source texts, and examine human and machine translations in order to learn to predict whether the machine translation of the first noun must replace the second one, or vice-versa, or no change should be made. The proposed classifiers use a variety of lexical, syntactic and semantic features. When presented with previously unseen source texts and baseline MT output, the decisions of the classifiers serve to post-edit or re-rank the repeated nouns of the MT baseline. This original end-to-end MT system generates improved Chinese-to-English and German-to-English translations, with larger improvements on the latter pair. Syntactic features appear to be more useful than semantic ones, for reasons that will be discussed. The combined re-ranking and post-editing approach appears to be the most effective one, converting about 50% of the gap in BLEU scores between the baseline MT and the use of an oracle classifier. The case of more than two consecutive occurrences of the same noun is briefly examined at the end of Chapter 5.

1.1.2 Word Sense Disambiguation for MT

In the second part of the thesis, we design a sense-aware MT system that attempts to automatically select the correct translation of ambiguous words by considering the sense knowledge during the translation process. Indeed, selecting the correct translation of polysemous words remains an important challenge for MT. Current statistical or neural MT systems perform word sense disambiguation (WSD) implicitly, for instance through the n-gram frequency information stored in the translation and language models. However, the context taken into account by an MT system when performing implicit WSD is limited. For instance, in the case of phrase-based SMT, it is related to the order of the language model (often between 3 and 5) and the length of n-grams in the phrase table (seldom above 5). In attention-based neural MT systems, the context extends to the entire sentence, but is not specifically trained to be used for WSD.

In Chapter 7, we introduce several adaptive WSD systems to be used for MT, which consider a larger context than is accessible to state-of-the-art MT systems. Our WSD system performs context-dependent clustering of word occurrences and is initialized with knowledge from WordNet, in the form of vector representations of definitions or examples for each sense. Later on, the labels of the resulting clusters are used as abstract source-side sense labels within a factored phrase-based SMT system.

The results, presented in Chapter 8, show first that our WSD system is competitive on the SemEval 2010 WSD task, but especially that it helps the translation of polysemous nouns and verbs, when translating from English into Chinese, German, French, Spanish or Dutch, in comparison to an SMT baseline that is not aware of word senses.

As the sense integration appears promising for SMT, we also transfer this approach to the newer neural MT models, which are now state of the art. In attention-based neural MT, the context extends to the entire sentence, but multiple word senses are not modeled explicitly. The implicit sense information captured by word representations used in NMT leads to a bias in the attention mechanism towards the dominant senses. Therefore, the NMT decoders cannot clearly identify the contexts in which one word sense should be used instead of another one. Hence, while NMT can use local constraints to translate “great rock band” into French as “*superbe groupe de rock*” rather than “*grande bande de pierre*” – correctly assigning to “rock" its musical rather than geological sense – it fails to do so for word senses which require larger contexts.

However, unlike SMT, for which it is easier to use linguistic features, NMT uses vectors for word generation, so traditional feature incorporation does not work here. In Chapter 9, we show how the explicit modeling of word senses can be helpful to NMT, by using combined vector representations of word types and senses, which are inferred from contexts that are larger than that of state-of-the-art NMT systems.

Specifically, we integrate into NMT weakly supervised WSD approaches, based on adaptive clustering and operating on large word contexts. We experiment with three sense selection mechanisms for integrating WSD into NMT, respectively based on top, average, and weighted average (i.e., attention) of word senses. The best methods again show consistent improvements against baseline NMT on five language pairs: from English into Chinese, German, French, Spanish and Dutch.

1.2 Relation to the MODERN Research Project

The present thesis benefited from the framework of the MODERN SNSF Sinergia project: Modeling Discourse Entities and Relations for Coherent Machine Translation.[1] The MODERN project builds upon work in a previous Sinergia project (COMTIS: Improving the Coherence of Machine Translation Output by Modeling Inter-sentential Relations[2]) and focuses on lexical consistency in translation at the document level, for noun phrases, pronouns and other means by which reference to entities in discourse is established. The MODERN project was a collaboration between the Idiap Research Institute and the Universities of Zürich, Geneva, and Utrecht (the Netherlands). Idiap was the head of the project and was responsible for natural language processing and machine learning methods, while the University of Zürich developed MT models incorporating lexical consistency and semantic ontologies. Linguistic issues, such as coherence and readability of translation were studied at the University of Utrecht, e.g. with eye-tracking methods.

The studies constituting this thesis have been partly conducted in collaboration with Laura Mascarell, PhD student at the University of Zürich and member of the MODERN project as well. In the first part of this thesis, the design of the compound translation consistency method was based on an initial idea proposed at the University of Zürich [Mascarell et al., 2014]. Later on, in the work on translating repeated words, we collaborated with Laura Mascarell for feature extraction: more precisely, we experimented in detail with the syntactic features, while our colleague worked on semantic ones. We then merged all features and experimented jointly over the two language pairs. Finally, the various solutions for using consistency predictions in combination with MT are a specific contribution of this thesis.

The studies presented in this thesis have resulted in the following publications (the complete list appears in the CV at the end of the thesis):

- The method to improve the translation of compound pairs, described in Chapter 4, has been published as a conference paper to the ACL-IJCNLP Student Research Workshop [Pu et al., 2015].
- The work on improving translation by determining the consistent translations of repeated nouns, presented in Chapter 5, has been published as a conference paper at EACL [Pu et al., 2017a].
- The work on sense-aware SMT using the output of adaptive WSD clustering methods, described in Chapters 7 and 8, has been published as a conference paper at WMT [Pu et al., 2017b].

[1] See www.idiap.ch/project/modern, supported by the Swiss National Science Foundation under grant number 147653 between 2013 and 2017.

[2] See www.idiap.ch/project/comtis, supported by the SNSF under grant number 127510, 2010–2013.

- Chapter 9 has been submitted in spring 2018 as a journal paper to the Transactions of the Association for Computational Linguistics (TACL), has been revised and resubmitted at the time of writing.

2 Background

This chapter provides an introduction with mathematical definitions to the field of statistical and neural machine translation (Section 2.1) which are the basis of our experiments. Later on, we present several approaches that improve machine translation in several directions, including lexical consistency with cache-based models, translation consistency based on anaphora resolution, and others (Section 2.2).

2.1 Introduction to Machine Translation Models

How do computers translate texts from one language to another? Human translators use a large amount of detailed knowledge about how the world works to correctly translate all the different meanings the same word or phrase can have in different contexts. This makes automated translation a hard problem, the kind of problem that may require genuine artificial intelligence.

2.1.1 Statistical Machine Translation

The phrase-based model is the most widely used one in statistical machine translation systems; this model translates small word sequences at a time. This section explains the basic principles of phrase-based models and how they are trained, and takes a more detailed look to extensions of the main components: the translation model, the language model and the reordering model, by following the explanations provided by Koehn [2009].

Statistical machine translation starts with a large data set of human translations, that is, a corpus of texts (e.g., United Nations documents in some of our studies) which have already been translated into multiple languages, and then uses those texts to automatically infer a statistical model of translation. That statistical model is then applied to new texts to make a guess for a reasonable translation.

Mathematical definition

Imagine you are given a foreign text **f**, and you would like to find a good English translation **e**. There are many possible translations of **f** into English and different translators will have different opinions about what the best translation **e** is. We can model these differences of opinion with a probability distribution $p(\mathbf{e}|\mathbf{f})$ over possible translations **e**, given that the foreign text was **f**. A reasonable way of choosing the "best" translation is to choose **e** which maximizes the conditional probability $p(\mathbf{e}|\mathbf{f})$.

Hence, the best English translation $\mathbf{e}_{best}$ of a foreign input sentence **f** is defined as

$$\begin{aligned}\mathbf{e}_{best} &= argmax_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) \\ &= argmax_{\mathbf{e}} p(\mathbf{f}|\mathbf{e})\, p_{LM}(\mathbf{e})\end{aligned}$$

Here the formula applies the Bayes rule to invert the translation direction and integrate a language model p_{LM}. For the phrase-based model, we decompose $p(\mathbf{f}|\mathbf{e})$ further into

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\, d(start_i - end_{i-1} - 1) \tag{2.1}$$

The foreign sentence **f** is broken up into I phrases $\bar{f}_i$. Note that this process of segmentation is not modeled explicitly. This means that any segmentation is equally likely.

Each foreign phrase $\bar{f}_i$ is translated into an English phrase $\bar{e}_i$. Since we mathematically inverted the translation direction in the noisy channel, the phrase translation probability $\phi(\bar{f}_i|\bar{e}_i)$ is modeled as a translation from English to the foreign language.

Reordering model

Reordering is handled by a distance-based reordering model. The model considers reordering relative to the previous phrase. It defines $start_i$ as the position of the first word of the foreign input phrase that translates to the ith English phrase, and end_i as the position of the last word of that foreign phrase. The reordering distance is computed as $start_i - end_{i-1} - 1$.

The reordering distance is the number of words skipped (either forward or backward) when taking foreign words out of sequence. If two phrases are translated in sequence, then $start_i = end_{i-1} + 1$; i.e., the position of the first word of phrase i is the same as the position of the last word of the previous phrase plus one. In this case, a reordering cost of $d(0)$ is applied.

What is the probability of d? Instead of estimating reordering probabilities from data, the model applies an exponentially decaying cost function $d(x) = \alpha^{|x|}$ with an appropriate value for the parameter $\alpha \in [0,1]$ so that d is a proper probability distribution. This formula simply means that the movements of phrases over large distances are more expensive than shorter movements or no movement at all.

Language model

In the language model, our goal is to find a tractable representation of the function $p(e_1^I)$. Generative models often use probabilistic tools for this. One of these is the chain rule, which allows us to write the following:

$$p(e_1^I) = \prod_{j=1}^{I} p(e_i|e_1^{i-1}) \tag{2.2}$$

Equation 2.2 tells us that the conditional probability of the sentence e_1^I is simply the product of many small probabilities, each of which corresponds to a single word.

Equation 2.2 helps to simplify our problem, but not completely. For instance, the distribution $p(e_I|e_1^{I-1})$ assigned to the last word of the sentence contains nearly as many terms as $p(e_1^I)$ itself. In order to simplify the model even further we introduce the idea of conditional independence, meaning that $p(x|y) = p(x)$. In other words, conditional independence means that knowing the value of y does not affect the probability distribution of x. By making independence assumption about our data, we can drop enough terms from our functions so that they become tractable.

In language modeling, the simplest assumption we can make is that the probability of word e_i is conditionally independent of all but the $n-1$ preceding words e_{i-n}^{i-1}. We call e_{i-n}^{i} an *n-gram* and the language model based on this independence assumption is an *n-gram language model*. We assume that the first word e_1 is preceded by $n-1$ "start symbols" not in V_E.

So using such notation we can rewrite Equation 2.2 as an n-gram model:

$$p(e_1^I) = \prod_{j=1}^{I} p(e_i|e_1^{i-1}) = \prod p(e_i|e_{i-n}^{i-1}) \tag{2.3}$$

Translation model

The standard model described so far consists of three factors:

- the phrase translation table $\phi(\bar{f}|\bar{e})$;
- the reordering model d;
- the language model $p_{LM}(e)$.

These three model components are multiplied together to form the phrase-based statistical machine translation model:

$$e_{best} = argmax_e \prod_{i}^{I} \phi(\bar{f}_i|\bar{e}_i) d(start_i - end_{i-1} - 1) \prod_{i=1}^{|e|} p_{LM}(e_i|e_1...e_{i-1}) \tag{2.4}$$

Another way to describe this is that there are three components that contribute to producing the best possible translation, by ensuring that

- the foreign phrases match the English words (ϕ);
- phrases are reordered appropriately (d);
- the output is fluent English (p_{LM}).

When we use our system, we may observe that the words between input and output match up pretty well, but that the output is not very good English. Hence, we are inclined to give the language model more weight. Formally, we can do this by introducing **weights** λ_{ϕ}, λ_d, λ_{LM} that let us scale the contributions of each of the three components:

$$e_{best} = argmax_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi} d(start_i - end_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|e|} p_{LM}(e_i|e_1...e_{i-1})^{\lambda_{LM}} \tag{2.5}$$

The assumption behind the translation model that the translation of a phrase does not depend on surrounding phrases is such a necessary but inaccurate assumption. Similarly, the tri-gram language model assumption states that the probability of an English word depends only on a window of two previous words. It is possible to find counterexamples for either of these assumptions.

By adding weights, we are guided more by practical concerns than by mathematical reasons. However, we do come up with a model structure that is well-known in the machine learning community: a log-linear model. Log-linear models have the following form:

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x) \tag{2.6}$$

Equation 2.5 fits this form with

- number of feature functions $n = 3$;

- random variable $x = (e, f, start_i, end_{i-1})$;
- feature function $h_1 = \log \phi$;
- feature function $h_2 = \log d$;
- feature function $h_3 = \log p_{LM}$.

To make this more apparent, here is reformulation of Equation 2.5:

$$
\begin{aligned}
p(e, a|f) = \exp \big[& \lambda_\phi \sum_{i=1}^{I} \log \phi(\bar{f}_i|\bar{e}_i) \\
& + \phi_d \sum_{i=1}^{I} \log d(start_i - end_{i-1} - 1) \\
& + \phi_{LM} \sum_{i=1}^{|e|} \log p_{LM}(e_i|e_1 ... e_{i-1}) \big]
\end{aligned}
\tag{2.7}
$$

Log-linear models are widely used in the machine learning community. For instance, Naive Bayes, maximum entropy, and perceptron learning methods are all based on log-linear models.

In this framework, we view each data point (a sentence translation) as a vector of features and the model as a set of corresponding feature functions. The feature functions are trained separately, and combined assuming that they are independent of each other.

The original justification for decomposing the model into a translation model and a language model was the noisy-channel model. We applied the Bayes rule, which is a mathematically correct transformation. However, we followed that up with a number of independence assumptions that are not strictly correct, but are necessary to decompose the model further into probability distributions for which we have sufficient statistics.

Decoding

After the training step, the parameters of the translation model can be tuned, usually by so-called Minimum Error Rate Training (MERT) [Och, 2003]. The MERT algorithm optimizes linear weights relative to n-best lists of possible translations generated from a separate development (or tuning) corpus. The latter is much smaller than the training corpus and only consists of a few thousand sentences. The randomized optimization iterates between optimizing weights and re-decoding with those weights to enhance the approximation to the best translation. Optimization is usually based on a loss function and for SMT, this is most often the BLEU evaluation metric.

At testing time, the so-called "decoding" is the construction of the output sentence as a sequence from left to right by incrementally computing the sentence translation probability with the mentioned feature scores in the phrase table and formula shown above. For decoding, a beam search including stacking, hypothesis expansion and pruning is run over the phrase translation table in order to guarantee computability and performance. A proper trade-off between speed (small beam size) and performance (large beam size) has to be found.

2.1.2 Factored Translation Model

The factored translation model [Koehn and Hoang, 2007] has been proposed as a general way to use additional knowledge within the SMT paradigm, possibly coming from text-level features. Currently it is implemented in SMT toolkits such as Moses [Koehn et al., 2007]. The factors are most often used to add morphological information or semantic information.

In its mathematical definition, as for the phrase-based model, the factored translation model can be seen as the combination of several components (language model, reordering model, translation steps, generation steps). These components define one or more feature functions that are combined in a log-linear model:

$$p(\vec{e}|\vec{f}) = \frac{1}{Z}\exp\sum_{i=1}^{n}\lambda_i h_i(\vec{e},\vec{f}) \tag{2.8}$$

where Z is a normalization constant that is ignored in practice. To compute the probability of a translation $\vec{e}$ given an input sentence $\vec{f}$, we have to evaluate each feature function h_i. For instance, the feature function for a bi-gram language model component is:

$$\begin{aligned} h_{LM}(\vec{e},\vec{f}) &= p_{LM}(\vec{e}) \qquad (2.9)\\ &= p(e_1)p(e_2|e_1)...p(e_m|e_{m-1}) \qquad (2.10)\end{aligned}$$

where m is the number of words e_i in the sentence $\vec{e}$.

Let us now consider the feature functions introduced by the translation and generation steps of factored translation models. The translation of the input sentence f into the output sentence e breaks down to a set of phrase translations $(\bar{f}_j, \bar{e}_j)$. For the translation step component, each feature function h_T is defined over the phrase pairs $(\bar{f}_j, \bar{e}_j)$ given a scoring function τ:

$$h_T(\vec{e},\vec{f}) = \sum_j \tau(\bar{f}_j, \bar{e}_j) \tag{2.11}$$

For the generation step component, each feature function h_G given a scoring function γ is defined over the output words e_k:

$$h_G(\vec{e}, \vec{f}) = \sum_k \gamma(e_k) \tag{2.12}$$

The feature functions follow from the scoring function (τ, γ) acquired during the training of translation and generation tables. The feature weights λ_i in the log-linear model are determined using a minimum error rate training method such as MERT:

$$f = \arg\max_f \exp \sum_{i=1}^{n} \lambda_i h_i(e, f) \tag{2.13}$$

where n is the number of features, $h_i(e, f)$ is the feature function over factors, and λ_i is the weight for that factor, which is optimized during tuning.

Several studies integrated additional linguistic knowledge with machine translation using factored-based MT models. Birch et al. [2007] used supertags from a Combinatorial Categorical Grammar as factors in phrase-based translation model. When the supertags (combined with other factors, e.g. POS tags) were applied to the target language side only, the factored models slightly improved over a phrase-based model for Dutch-English translation. Avramidis and Koehn [2008] added source-side syntactic information for each word for translating from a morphologically poorer language to a richer one (English-Greek). They achieved 0.15-0.50 BLEU improvement.

2.1.3 Neural Machine Translation

Attention-based NMT models [Bahdanau et al., 2015, Luong et al., 2015] have become the state-of-the-art in machine translation, improving over the initial sequence-to-sequence models using deep neural networks [Cho et al., 2014, Sutskever et al., 2014].

Given a source sentence X with words w^x, $X = (w^x_1, w^x_2, ..., w^x_T)$, the attention-based NMT model computes a conditional distribution over translations $p(Y = (w^y_1, ..., w^y_{T'})|X)$. The neural network model consists of an encoder, a decoder and an attention mechanism.

Encoder

The encoder is implemented as a bi-directional recurrent neural network that reads the source sentence word-by-word. Before being read by the encoder, each source word $w^x_t \in V$ is

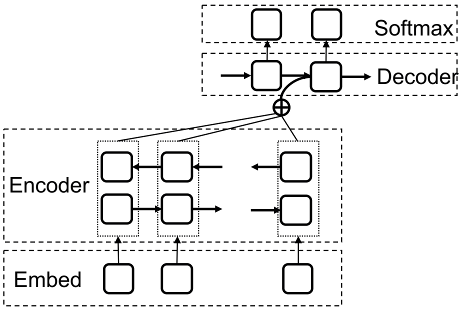


Figure 2.1 – Attention-based neural MT model used as a baseline.

projected from a one-hot word vector into a continuous vector space representation $\vec{x}_t$.

$$\vec{x}_t = E^{\vec{x}} 1(w_t^x) \tag{2.14}$$

where $\vec{1}(w_t^x)$ is a one-hot vector defined as

$$\vec{1}(w_t^x)_j \propto \begin{cases} 1 & \text{if } j = w_t^x \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

The resulting sequence of word embedding vectors is then read by the bidirectional encoder recurrent network which consists of forward and backward recurrent networks. The forward recurrent network reads the sequence in the left-to-right order, i.e. $\overrightarrow{\vec{h}}_t = \overrightarrow{\phi}(\overrightarrow{\vec{h}}_{t-1}, \vec{x}_t)$, while the backward network reads it right-to-left: $\overleftarrow{\vec{h}}_t = \overleftarrow{\phi}(\overleftarrow{\vec{h}}_{t+1}, \vec{x}_t)$.

The hidden states from the forward and backward recurrent networks are concatenated at each time step t to form an 'annotation' vector $\vec{h}_t = [\overrightarrow{\vec{h}_t}; \overleftarrow{\vec{h}_t}]$. Taken over several time steps, these vectors form the 'context' i.e. a tuple of annotation vectors $C = (\vec{h}_1, \vec{h}_2, ..., \vec{h}_T)$. The recurrent activation functions $\overrightarrow{\phi}$ and $\overleftarrow{\phi}$ are either long short-term memory units (LSTM) or gated recurrent units (GRU).

Decoder

The decoder recurrent network maintains an internal hidden state $z_{t'}$. After each time step t', it first uses the attention mechanism to select, or weight, the annotation vectors in the context tuple C. The attention mechanism takes as input both the previous decoder hidden state, and one of the annotation vectors, and returns a relevance score $e_{t',t} = f_{\text{ATT}}(z_{\vec{t'}-1}, \vec{h}_t)$. These scores are normalized to obtain attention scores, as follows:

$$\alpha_{t',t} = exp(e_{t',t}) / \sum_{k=1}^{T} exp(e_{t',k}) \tag{2.16}$$

The normalized attention scores serve to compute a weighted sum of annotation vectors, which will be used by the decoder's recurrent network to update its own hidden state:

$$\vec{c}_{t'} = \sum_{t=1}^{T} \alpha_{t',t} \vec{h}_t \text{ and } z_{t'} = \phi_z(z_{t'-1}, \vec{y}_{t'-1}, \vec{c}_{t'}) \tag{2.17}$$

Similarly to the encoder, ϕ_z is implemented using either LSTMs or GRUs and $\vec{y}_{t'-1}$ is the target-side word embedding vector corresponding to word w^y, computed as $\vec{y}_{t'-1} = \vec{E}^y \vec{1}(w^y_{t'-1})$, similarly to Eq. 2.14. The probability of each word i in the target vocabulary V' is computed by

$$p(w^y_{t'} = i | w^y_{<t'}, X) \propto exp(\vec{E}^y_i z_{t'} + c_i)$$

where $\vec{E}^y_i$ is the i-th row vector of the target word embedding matrix.

The NMT model is trained to maximize the log-probability of the correct translation given a source sentence using a large parallel corpus. The training is done by stochastic gradient descent, where the gradient of the log-likelihood is efficiently computed by the back-propagation algorithm.

2.2 Research on Discourse and MT

State-of-the-art machine translation (MT) systems, especially statistical but also neural ones, operate in a sentence-by-sentence mode, and do not propagate information through the series of sentences that constitute a complete text. While this reduces the complexity of the translation process, it introduces the problem that important discourse-level information is lost, such as antecedent/pronoun and co-reference links, which require context provided by previous sentences. The propagation of such information is indispensable to make correct translation choices for words and phrases that depend on it. The aim of our research is to model and automatically detect such dependencies, and to study their integration within MT with the aim of demonstrating improvement in translation quality. Below, we present several solutions that have been proposed to use text-level dependency information for MT.

In other studies, factored translation models [Koehn and Hoang, 2007] have been used with the same purpose as in ours, for instance by incorporating contextual information into labels used to indicate the meaning of ambiguous discourse connectives [Meyer and Popescu-Belis, 2012] or the expected tenses of verb phrase translations [Loaiciga et al., 2014]. Quite naturally, there are analogies between our work and studies of pronoun translation [Le Nagard and Koehn, 2010, Hardmeier and Federico, 2010a, Guillou, 2012], with the notable difference that pronominal anaphora resolution remains a challenging task. Our work also contributes to the general objective of using discourse-level information to improve MT [Hardmeier, 2014, Meyer, 2014].

Several studies tried to improve translation consistency by memorizing the previous translation through cache-based models. Tiedemann [2010] used cache-based approaches to phrase-based SMT on out-of-domain data. Their main proposal was to test the use of adaptive language and translation models in a standard phrase-based SMT setting for the exploitation of a wider context, beyond sentence boundaries. A cache-based language model mixes a large global language model with a small local model estimated from recent items in the history of the input stream. The cache stores the translated path (or phrase table) which is used during the decoding process and increases the probability of such choices, so that later they have more chances to use the same pairs. This model was designed to improve the consistency of the MT output during decoding, and obtained a gain of about 2.6% relative BLEU points per sentence. From these results, it appears that cache-based models positively impact the result on in-domain test sets, but not on out-of-domain ones.

Pronominal anaphora is the use of a pronoun to refer to an entity mentioned earlier in the discourse. This happens frequently in most types of connected text. Le Nagard and Koehn [2010] approached the pronoun translation problem in phrase-based SMT by processing documents in two passes. The English input text was run through a co-reference resolver, and translation was performed with a regular SMT system to obtain French translations of the antecedent noun phrase. Then the anaphoric pronouns of the English text were annotated with the gender and number of the French translation of their antecedent and translated again with another MT system whose phrase tables had been annotated in the same way. This did not result in any noticeable increase in translation quality, a fact that the authors attributed to the insufficient quality of their co-reference resolution system. However, in a later application of the same approach to an English-Czech system, no clearly positive results were obtained despite the use of manually annotated data for co-reference [Guillou, 2012].

Hardmeier and Federico [2010b] studied the pronoun translation problem in a phrase-based SMT system that directly incorporates the processing of co-reference links into the decoding step. Pronoun co-reference links were annotated with the BART software. The authors then added an extra feature to the decoder to model the probability of a pronoun given its antecedent. Sentence-internal co-reference links were handled completely within the SMT dynamic programming algorithm. In this work, no improvement in BLEU score was achieved for English-German translation, but a slight improvement was found with an evaluation metric targeted specifically to pronoun co-reference.

From the results presented above it can be seen that attempts at improving translation at the sentence-level by simply enforcing consistent vocabulary choices are quite limited, since the SMT vocabulary is already fairly consistent, which is due to the fact that overall lexical consistency is “natural” to SMT, but that the correct translation of NPs referring to entities is still problematic. The cache-based approach adds one more information about the previous translations of words, which lets the system generate the translation of current word tokens more consistently. Similarly, the studies that attempt to improve pronoun translation with the help of anaphora resolution can also be considered as attempts to improve the consistency of translations. In our work, we do not only analyze pronouns, but consider all word occurrences in a document, including those that could help anaphora resolution for MT as well.

2.3 Evaluation Metrics

Automatic scoring of translation quality is a difficult problem and has become a research task in its own right over the years. This is mainly due to the fact that there is no single, one-best translation and that human reference translations differ considerably across translators even for short sentences.

The automatic metrics most often referred to in the literature all rely on the same scoring principle: the overlap of a system's output (or candidate translation) with one human reference translation, or depending on availability, several different reference translations. This overlap can be measured by various approaches: the BLEU score [Papineni et al., 2002] for example, counts overlap in terms of matching n-grams, and is the most frequently used metric in the MT community. The more matches there are for 4-, 3-, 2- and 1-grams in a candidate translation compared to its reference, the higher the BLEU score. The values of the score range from 0 to 100, where 100 is reached for identical translations. State-of-the-art systems, depending on the language pairs involved, tend to have values between 10 and 40 BLEU points. Although frequently criticized for its limitations, BLEU remains a fast, language-independent and freely-available metric for MT, which correlates rather well with human judgments of translation quality, especially when averaged over a large quantity of text.

Other frequently used measures are METEOR and TER. The former considers possible word re-ordering and synonyms (with values similar to BLEU) and the latter computes a string edit distance, in terms of word insertion or deletion, measuring the effort that would needed to transform a candidate into a reference translation: the smaller this edit distance is, the better the translations. For our task, we most often compare a modified, discourse-aware MT system against a baseline system and report BLEU scores as the automatic method for evaluation.

In addition to automatic and objective evaluation, we also evaluate our models in a subjective way, i.e. where humans manually inspect the accuracy of each translated sentence. We present the translation outputs provided by both a baseline and our system, in a random order, and ask subjects to judge the translation quality of the two candidates. The subjects are asked to give a score for each translation based on the translation quality and we integrate scores from several subjects and present their distributions, with the goal of indicating whether our systems are judged as significantly better than baselines.

Part I

Lexical Consistency for Machine Translation

3 Previous Work on Consistency in SMT

Words tend to be less ambiguous when considered in larger contexts. In this part, we aim to reduce word ambiguity by using lexical consistency across sentences. Our study in this part is divided into two chapters. First, in Chapter 4, we correct the translation of polysemous words by constraining the translation of the head of a compound when it is repeated separately after it. The initial compound idea was first published by Mascarell et al. [2014], in which the coreference of compound noun phrases in German (e.g. Nordwand/Wand) was studied and used to improve German-to-French translation by assuming that the last constituent Y of the compound should have the same translation as that of Y in XY. We extend this work by designing compound detection rules for Chinese and use it to improve Chinese-to-English translation by forcing the detected compound pairs to be translated consistently. Second, the study in Chapter 5 is considerably more general, as it makes no assumption on either of the repeated nouns, i.e. it does not require them to be part of compounds.

The study in this part is related to several research topics in MT: lexical consistency, caching, co-reference, and long-range dependencies between words in general. Our proposal aims to improve the consistency of noun translation, and thus has a narrower scope than the “one translation per discourse” hypothesis [Itagaki et al., 2007, Carpuat, 2009, Carpuat and Simard, 2012], which aimed to implement for MT the broader hypothesis of “one sense per discourse” [Gale et al., 1992].

Several previous studies focused on enforcing consistent lexical choices. Tiedemann [2010] proposed a cache-model to enforce consistent translation of phrases across the document. However, caching is sensitive to error propagation, that is, when a phrase is incorrectly translated and cached, the model propagates the error to the following sentences. Gong et al. [2011] later extended Tiedemann’s proposal by initializing the cache with phrase pairs from similar documents at the beginning of the translation and by also applying a topic cache, which was introduced to deal with the error propagation issue. Xiao et al. [2011] defined a three-step procedure that enforces the consistent translation of ambiguous words, achieving improvement for English-to-Chinese MT. Ture et al. [2012] encouraged consistency for Arabic-to-English MT by introducing cross-sentence consistency features to the translation model, while Alexandrescu and Kirchhoff [2009] enforced similar translations over sentence having a similar graph representation.

Our first study in Chapter 4 is an instance of a recent trend aiming to go beyond sentence-by-sentence MT, by using semantic information from previous sentences to constrain or correct the decoding of the current one. In the study, we compare caching and post-editing as ways of achieving this goal. In other studies, factored translation models for SMT [Koehn and Hoang, 2007] have been used with the same purpose, by incorporating several kinds of linguistic information into SMT to improve translation quality. For instance, Meyer [2014] improved the translation of discourse connectives by integrating contextual information as additional features to SMT through a factored model, while Loaiciga et al. [2014] labeled the tenses of verb phrases and let the translation consider the tense information by using a factored model during the generation process.

As we discussed in Section 2.2, several researches tried to improve translation by considering the antecedents of pronouns. Several recent studies considered co-reference to improve pronoun resolution, but none of them successfully exploited noun phrase co-reference, likely due to an insufficient accuracy of co-reference resolution systems [Callin et al., 2015, Luong and Popescu-Belis, 2016, Werlen and Popescu-Belis, 2017]. The improvement of pronoun translation was only marginal with respect to a baseline SMT system in a 2015 shared task on pronoun translation [Hardmeier et al., 2015], while the 2016 shared task [Guillou et al., 2016] somewhat shifted its focus to pronoun prediction in lemmatized reference translations. Our work in Chapter 4 and its perspectives contribute to the more general objective of using discourse-level information to improve MT [Hardmeier, 2013, Loaiciga et al., 2014].

In the second study (Chapter 5), we extend our findings from Chapter 4 by focusing on repeated nouns. As indicated in Chapter 4, MT systems trained on small datasets are often more consistent but of lower quality than systems trained on larger and more diverse data sets. In our study, we do not alter consistent translations, but we address inconsistencies, which are often translation errors [Carpuat and Simard, 2012], and attempt to find those that can be corrected simply by enforcing consistency. However, our scope is narrower than the caching approach proposed by Tiedemann [2010], Gong et al. [2011], Bertoldi et al. [2013], which encourages consistent translations of any word, with the risk of propagating cached incorrect translations. In our study, the first and second translation in a pair have equal status.

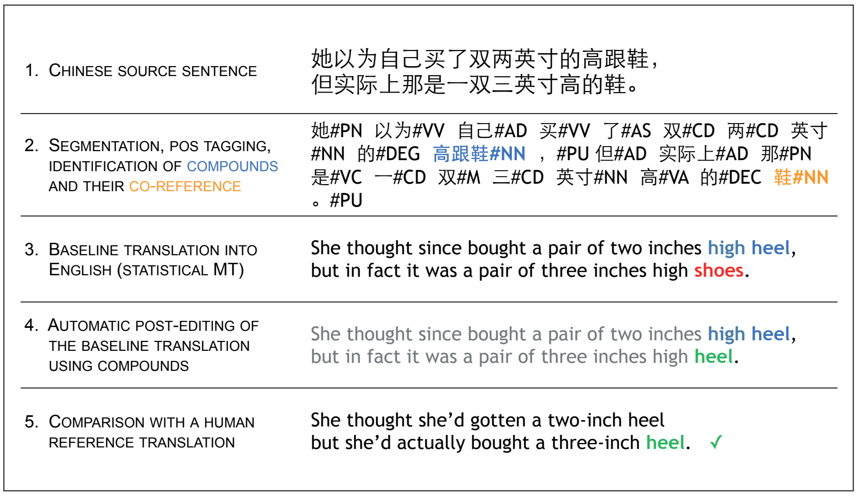
The second chapter in this part (Chapter 5) builds upon and extends the study of the first one (Chapter 4) on the translation of compounds [Mascarell et al., 2014, Pu et al., 2015]. This work is a collaboration with Laura Mascarell from the University of Zürich. Our role was mainly to experiment with the syntactic feature analysis while our colleague worked on semantic ones. We then merged all extracted features together and experimented jointly over two language pairs.

This work contributes to a growing corpus of research on modeling longer-range dependencies than those modeled in phrase-based SMT or neural MT, often across different sentences of a document. Ture et al. [2012] used cross-sentence consistency features in a translation model, while Hardmeier et al. [2012] designed the Docent decoder, which can use document-level features to improve the coherence across sentences of a translated document. Our classifier for repeated nouns outputs decisions that can serve as features in Docent, but as the frequency of repeated nouns in documents is quite low, we use here post-editing and/or re-ranking rather than Docent.

4 Leveraging Compounds to Improve Noun Phrase Translation

This chapter presents an automatic method for the correction of ambiguous translations by leveraging the translation of compound pairs. This study makes the assumption that the translation of a noun-noun compound, noted XY, displays fewer ambiguities than the separate translations of its components X and Y. Therefore, on a subsequent occurrence of the head Y, assuming that Y refers to the same entity as XY, we hypothesize that the previously-found translation of Y within the XY compound is a better and more coherent translation than the one proposed for Y, when an SMT system is not aware of the compound.

For instance, in the first line of the example in Figure 4.1, the Chinese compound 高跟鞋 refers to 'high heels' and the subsequent mention of the referent using only the third character '鞋' should be translated as 'heels'. However, the character '鞋' by itself could also be translated as 'shoe' or 'footwear', as observed with a baseline SMT system that is not aware of the XY/Y coreference.



1. Chinese source sentence	她以为自己买了双两英寸的高跟鞋， 但实际上那是一双三英寸高的鞋。
2. Segmentation, POS tagging, identification of compounds and their co-reference	她#PN 以为#VV 自己#AD 买#VV 了#AS 双#CD 两#CD 英寸#NN 的#DEG 高跟鞋#NN ，#PU 但#AD 实际上#AD 那#PN 是#VC 一#CD 双#M 三#CD 英寸#NN 高#VA 的#DEC 鞋#NN 。#PU
3. Baseline translation into English (statistical MT)	She thought since bought a pair of two inches high heel, but in fact it was a pair of three inches high shoes.
4. Automatic post-editing of the baseline translation using compounds	She thought since bought a pair of two inches high heel, but in fact it was a pair of three inches high heel.
5. Comparison with a human reference translation	She thought she'd gotten a two-inch heel but she'd actually bought a three-inch heel. ✓

Figure 4.1 – Compound post-editing method illustrated on ZH/EN (excerpt from a 1998 TED talk by Aimee Mullins, *Changing my legs – and my mindset*, at 19:09).

Our claim is supported by results from experiments on Chinese-to-English and German-to-French corpus translation presented in Section 4.2. We demonstrate that the translations of nouns are indeed improved, as measured by automatic or human comparisons with the reference translation. Although the XY/Y occurrences may not appear very frequently in texts, the translation mistakes do make people misunderstand the text, since they often hide the co-reference link between two expressions.

4.1 Description of the Method

When an entity is referred to by a compound, a subsequent reference to it might use only the head of the constituent, to avoid repeating the entire compound. Figure 4.2 shows an example of compound pair translated by an SMT baseline system: the compound '蔬菜' (meaning vegetable), formed by '蔬' (X) and '菜' (Y) can be followed by '菜' (Y) alone to refer again to the same entity. In such cases, the appropriate meaning of the noun Y (among its multiple possible ones) must be identified, typically by assuming that the text is consistent and that Y and XY co-refer. When applying these considerations to machine translation, the main aim of our method is to detect such cases, and to enforce that the head of XY, which is the second constituent Y in Chinese or German, must have the same translation for both XY and Y.

Example:
Source: 这是一种中国特有的蔬菜，这种菜含有丰富的维他命。
Human Translation: This is a special kind of Chinese vegetables, these vegetables are rich in vitamins.
SMT: This is a unique Chinese vegetables, this dish is rich in vitamins.

Figure 4.2 – Inconsistent translations of a compound pair, in blue, from Chinese into English. Human consistently translates the compound pair while the SMT baseline translates Y as 'dish', which is a mistake.

4.1.1 Overview

We propose to use the translation of a compound XY to improve the translation of a subsequent occurrence of Y, the head of the XY noun phrase, in the following way, represented schematically in Figure 4.1. Details for each stage are given below in Sections 4.1.2 and 4.1.3.

First, the presence of XY/Y patterns is detected either by examining whether a compound XY is followed by an occurrence of Y, or, conversely, by examining for each Y candidate whether it appears as part of a previous compound XY. Distance constraints and additional filtering rules are implemented to increase the likelihood that XY and Y are actually co-referent, or at least refer to entities of the same type.

Second, each sentence is translated by a baseline SMT system, and the translation of the head Y of each compound XY is identified using the word alignment from the SMT decoder. This translation is used as the translation of a subsequent occurrence of Y either by caching the corresponding source/target word pair in the SMT or by post-editing the baseline SMT output.

For instance, if the Chinese pair (蔬菜, 菜) shown in Figure 4.2 is identified, where the first compound can unambiguously be translated into English by ‘vegetable’, then the translation of a subsequent occurrence of ‘菜’ is enforced to ‘vegetable’. This has the potential to improve over the baseline translation, because when considered individually, ‘菜’ could also be translated as ‘dish’, ‘greens’, ‘wild herbs’, etc.

4.1.2 Identifying XY/Y Pairs

Chinese and German share a number of similarities regarding compounds. Although Chinese texts are not word-segmented, once this operation is performed, multi-character words in which all characters have individual meanings – such as the above-mentioned ‘蔬菜’ meaning ‘vegetable’ – are frequent. Similarly, in German, noun-noun compounds such as ‘Bundesamt’ (‘Bund’ + ‘Amt’, for Federal Bureau) or Nordwand (‘Nord’ + ‘Wand’, for North face) are frequent as well. While the identification of XY noun-noun compounds is straightforward with morpho-syntactic analysis tools, the identification of a subsequent mention of the head noun, Y, and especially the decision whether this Y refers or not to the same entity XY, are more challenging issues. In other words, the main difficulty is to separate true XY/Y pairs from false positives.

To increase the chances that XY/Y pairs are indeed co-referent, we narrow down the set of detected cases using hand-written rules that check the local context of Y. For example, we consider only the cases where Y is preceded by demonstrative pronouns (e.g. ‘这’ or ‘那’ meaning ‘this’ and ‘that’ in Chinese, or ‘diese’ in German), possessive pronouns, and determiners (‘der’, ‘die’, ‘das’ in German). Since other words can occur between the two parts (like classifiers[1] in Chinese or adjectives), there are additional distance constraints: the pronoun or determiner must be separated by fewer than three words. Since the rules use morphological information and word boundaries, they are preceded by word segmentation[2] and tagging[3] for Chinese and morphological analysis for German.[4] For example, in the input sentence from Figure 4.1, we determine that the noun phrase ‘鞋’ fits our condition for extraction as Y because there are words before it which fulfil the condition for acceptance.

Specifically, our selection rules for Chinese are the following ones:

- Singular pattern: 这 or 那 + classifier + Y with distance constraints.
- Plural condition: 这些 or 那些 (meaning ’these’ and ’those’) + Y with distance con-

[1] Classifier words in Chinese are mainly inserted between a numeral and the noun qualified by it, such as “one person” or “three books”.

[2] Using the Stanford Word Segmenter available from http://nlp.stanford.edu/software/segmenter.shtml.

[3] Using the Stanford Log-linear Part-of-speech Tagger, http://nlp.stanford.edu/software/tagger.shtml.

[4] Using Gertwol [Koskeniemmi and Haapalainen, 1994].

straints.

A classifier is a special term in Chinese language. The basic uses of classifiers are in phrases in which a noun is qualified by a numeral. When a phrase such as "one person" or "three books" is translated into Chinese, it is normally necessary to insert an appropriate classifier between the numeral and the noun. Since our goal is to correct the translation of Y, so here we only detect the classifiers which precede Y.

Moreover, we set a distance constraint, and select only the cases in which the distance between the classifier and the demonstratives '这' or '那' is smaller than three words, with Y being the first or second noun after the classifier, in the singular case.

Here we explain the process using the sample text shown in Figure 4.1:

- In the input sentence, we determine that the noun phrase 鞋 fits our condition for extraction as Y, and there exist words before 鞋 which fulfil the condition for acceptance, in the singular case: 那 (that) and 双 (classifier), and 鞋 follows these two keywords within acceptable distance (鞋 is the first noun phrase which appears after 双), so here we extract 鞋 as the possible Y in a pair XY/Y structure.
- After determining Y, we search for the noun phrases in the three previous sentences which have the structure XY. In the example in Figure 4.1, the noun phrase 高跟鞋 (in blue) satisfies this condition. Therefore, the pair (高跟鞋 , 鞋) is a Chinese compound for which the translation of both occurrences of Y should be consistent.

For German as a source language, which was studied by Laura Mascarell at the University of Zürich, the processing steps described in [Mascarell et al., 2014].

4.1.3 Enforcing the Translation of Y

Two language-independent methods have been designed to ensure that the translations of XY and Y are consistent: post-editing and caching. The second one builds upon an earlier proposal tested only on DE/FR with subjective evaluations [Mascarell et al., 2014].

In the post-editing method, we consider correcting the baseline translation of Y through the following steps:

- For each XY/Y pair, the translations of XY and Y by a baseline SMT system (described in Section 4.1.4) are first identified through word alignment based on IBM-3 alignment model.
- We verify if the baseline translations of Y in both noun phrases XY and Y are identical or different.

- If the translations are different, we replace the translation of Y by the translation of XY or by its head noun only, if it contains several words.

In the example in Figure 4.1, we first detect XY/Y pairs using the method explained in Section 4.1.2 and obtain the following compound followed by its head only: '高跟鞋' and '鞋'. Then we extract the baseline translations of the two noun phrases in the pair, which are respectively 'high heel' and 'shoes', by using the GIZA++ alignment tool. Since the translations are not identical, we replace the translation of Y ('shoes') by the translation of the head noun of XY ('heel'). Therefore, while the baseline SMT translated XY into 'high heel' and Y into 'shoes' (the latter being a wrong translation of '鞋' in this context), our method uses the consistency constraint and post-edits the translation of Y into 'heel', which is here the correct word.

Several differences from the ideal case presented above must be handled separately:

1. Several XY are candidates for co-reference with the same Y.
2. The compound XY is not translated at all (out-of-vocabulary source word).
3. The alignment of Y is empty in the target sentence (alignment error or untranslated word).

For condition 1, we address the issue via the additional constraints in the post-editing of Y:

- If the translations of the compounds XY which may be co-referent with a given Y are different, then we do not modify this Y, because we do not know which of the multiple preceding XY translations we should use. Moreover, we observed that simply using the most recent XY was not a reliable solution.
- If the translation of all XY which are co-referent with same Y are same, then we do not modify this Y, as well if the XY translation is only one word. We only change it if the translations consist of several words, ensuring that XY is translated with a compound noun phrase.

For condition 2, we actually use the translation of the Y noun as a translation of XY, instead of post-editing Y.

Finally, for condition 3, we actually consider the alignment of the word preceding Y, and if it is not empty, we use the translation of Y from the translation of XY to post-edit this word.

In the caching method [Mascarell et al., 2014], once an XY compound is identified, we obtain the translation of the Y part of the compound through the word alignment given by the SMT decoder. Next, we check that this translation appears as a translation of Y in the phrase table, and if so, we cache both Y and the obtained translation. We then enforce the cached translation every time a nous Y likely coreferent with XY is identified. This is different from the probabilistic caching proposed by Tiedemann [2010], because in our case the cached translation is deterministically enforced as the translation of Y.

4.1.4 Experimental Settings

The experiments are carried out on two different parallel corpora: the WIT^3 Chinese-English dataset [Cettolo et al., 2012] with transcripts of TED lectures and their translations, and the Text+Berg German-French corpus [Bubenhofer et al., 2013], a collection of articles from the yearbooks of the Swiss Alpine Club. The sizes of the subsets used for training, tuning and testing the SMT systems are given in Table 4.1. The test sets were constructed by selecting all the sentences and fragments that contain XY/Y pairs, identified as above, to maximize their number in the test data. These sentences are not needed in the training/tuning sets, as our proposal is not based on machine learning.

The rules for selecting co-referent XY/Y pairs in Chinese identified 261 pairs among 192k sentences. The rather low rate of occurrence (about one every 700 sentences) is explained by the strict conditions of the selection rules, which are designed to maximize the likelihood of coreference. In German, less restrictive rules selected 7,365 XY/Y pairs (a rate of one every 40 sentences). Still, in what follows, we randomly selected 261 XY/Y pairs from the DE/FR test data, to match their number in the ZH/EN test data.

Our baseline SMT system is the Moses phrase-based decoder [Koehn et al., 2007], trained over tokenized and true-cased data. The language models were built using SRILM [Stolcke et al., 2011] at order 3 (i.e. up to trigrams) using the default smoothing method (i.e. Good-Turing). Optimization was done using Minimum Error Rate Training [Och, 2003] as provided with Moses.

		Sentences	**Tokens**
ZH	Training	188’758	19’880’790
	Tuning	2’457	260’770
	Testing	855	12’344
DE	Training	285’877	5’194’622
	Tuning	1’557	32’649
	Testing	505	12’499

Table 4.1 – Sizes of the data sets for SMT.

4.2 Analysis of Results

The effectiveness of the proposed systems is measured in two ways. First, we use BLEU [Papineni et al., 2002] for overall evaluation, to check if our systems provide better translations for entire texts. Then, we focus on the XY/Y pairs and count the number of cases where the translations of Y match the reference or not, which can be computed automatically using the alignments from Moses.

4.2.1 Automatic Comparison with a Reference

The BLEU scores obtained by the baseline SMT, the caching and post-editing methods, and an oracle system are given in Table 4.2. The scores are in the same range as the baseline scores found by other teams [Cettolo et al., 2012] on the datasets shown in Table 4.1, and much higher on DE/FR than ZH/EN.

Our methods have a positive effect on ZH/EN translation, but a slightly negative effect on DE/FR one. Given the sparsity of XY/Y pairs with respect to the total number of words, hence the small number of changed words, these results meet our prior expectations. Indeed, we also computed the oracle BLEU scores for both language pairs, i.e. the scores when all Y members of XY/Y pairs are (manually) translated exactly as in the reference (last line of Table 4.2). These values are only slightly higher than the other scores, showing that even a perfect translation of the Y nouns would only have a small effect on BLEU.

	ZH/EN	DE/FR
BASELINE	11.18	27.65
CACHING	11.23	27.26
POST-EDITING	11.27	27.48
ORACLE	11.30	27.80

Table 4.2 – BLEU scores of the proposed methods.

We now turn to the reference-based evaluation of the translations of Y in the 261 XY/Y pairs, comparing the baseline SMT with each of our methods. These results are represented as four contingency tables – two language pairs and two methods against the baseline – gathered together as percentages in Table 4.3. Among these values, we first focus on the total number of pairs where our system agrees with the reference while the baseline system does not (i.e., improvements due to the system), and the converse case (degradations). The higher the difference between the two values, the more beneficial our method is.

For ZH/EN and the post-editing system, among the 222 extracted pairs, there were 45 pairs (20.3%) showing improvements of the system with respect to the baseline, and only 10 degradations (4.5%). There were also 94 pairs (42.3%) for which the baseline and the post-edited system were equal to the reference. The remaining 73 pairs (32.9%) will be analyzed manually in the next section. Therefore, from a pure reference-based view, the post-edited system has a net improvement of 15.8% (absolute) over the baseline in dealing with the XY/Y pairs.

A similar pattern is observed with the other method, namely caching, again on ZH/EN translation: 13.8% improvements vs. 4.1% degradations. The difference (i.e. the net improvement) is

			Caching		Post-editing	
			= ref	≠ ref	= ref	≠ ref
ZH/EN	Baseline	= ref	59.3	*4.1*	42.3	*4.5*
		≠ ref	**13.8**	22.8	**20.3**	32.9
DE/FR	Baseline	= ref	70.1	*10.3*	73.9	*5.0*
		≠ ref	**4.3**	15.2	**3.5**	17.5

Table 4.3 – Comparison of each approach with the baseline, for the two language pairs, in terms of Y nouns which are identical or different from a reference translation ('ref'). All scores are percentages of the totals. Numbers in **bold** are improvements over the baseline, while those in *italics* are degradations.

slightly smaller in this case with respect to the post-editing method.

For DE/FR translation, both methods appear to score fewer improvements than degradations. There are more than 70% of the pairs which are translated correctly by the baseline and by both systems, which indicates that the potential for improvement is much smaller for DE/FR than for ZH/EN.

While the pattern of improvement between ZH/EN and DE/FR is similar for post-editing and for caching, for both language pairs the post-editing method has a larger difference between improvements and degradations than the caching method. This can be explained by a lower coverage of the latter method, since it only enforces a translation when it appears as one of the translation candidates for Y in the phrase table [Mascarell et al., 2014].

We present several examples in Figure 4.3. In the first example, we correct the translation of Y by using the translation of XY and the new wording is identical to the reference. However, cases as the second and third examples in Figure 4.3 cannot be automatic evaluated, since the translations from both the baseline and our system are different from the reference. So measuring the effectiveness of the methods by automatic metrics is not sufficient. If a system's output is different from the reference, it can still represent an acceptable translation. For instance, in the Example 3 shown in Figure 4.3, both the baseline and post-edited translations differ from the reference, but the former translation, 'car', is not acceptable (the reference is 'bike') while the system's translation, 'bicycle', is fine. Therefore, for evaluating the method's usefulness, apart from automatic metrics, we conducted as well a human analysis, presented hereafter.

4.2.2 Subjective Evaluation of Undecided Cases in ZH/EN MT

In the human evaluation, all NP translations which differ from references are considered by three annotators. Each annotator puts the current translation into context (by looking at the previous sentences) to judge its quality on a three-point scale: good (score 2), acceptable (score 1) and bad (score 0). Finally, the consensus of all annotators is computed to evaluate

Example 1
Source: 在左边你能看到一个小船。这是一个15英寸的船。
Baseline: on the left , you can see a boat . this is a about 15 feet of the ship .
Post-editing: on the left , you can see a boat . this is a about 15 feet of boat .
Reference: You can see on the left side a small boat. That's about a 15-foot boat.

Example 2
Source: 在座的许多都知道这是贝多芬的第七交响曲。在座的许多都知道当他谱写这首曲子时，他已失聪，或近乎完全失聪。
Baseline: many of all know this is the seventh symphony. we know that when he dedicated this tune, he is deaf, or almost completely deaf
Post-editing: many of all know this is the seventh symphony. we know that when he dedicated this symphony, he is deaf, or almost completely deaf.
Reference: So many of you know that that's Beethoven's Seventh Symphony. We know that he was deaf, or near profoundly deaf, when he wrote that.

Example 3
Source: 你不能称它为设计自行车：这个车的签名是由钛制成的。
Baseline: you couldn't call this bicycle: designers designers by car is making.
Post-editing: you couldn't call this bicycle: designers designers by bicycle is making.
Reference: You wouldn't call this a designer bike: a designer bike is made of titanium.

Figure 4.3 – Translations of $X\,Y\,/\,Y$ pairs (blue) from Chinese into English: examples of improvements (green) made by our system with respect to an incorrect baseline (red).

the systems' performance.

Table 4.4 presents the confusion matrix that counts the number of translations of Y, from either the baseline or the post-editing systems, that are identical or not to the reference, on Chinese-to-English translation. As we explained above, we focus on the cases (49+24) in which the translation of both systems are not identical to the reference.

		POST-EDITING	
		= ref	≠ ref
Baseline	= ref	94	10
	≠ ref	45	49+24

Table 4.4 – Confusion matrix for our post-editing translation result against baseline over Chinese-to-English language pair.

As shown in Table 4.4, when both the baseline and the post-edited translations of Y differ from the reference, they can either be identical (49 cases) or different (24). In the former case, of course, neither of the systems outperforms the other, because their outputs are identical. The interesting observation is that the relative high number of such cases (49) is due to situations where the reference translation of noun Y is actually a pronoun (40), while MT systems have currently no possibility to generate a pronoun from a noun in the source sentence.

When both the baseline and one of our systems generate translations of Y which differ from the reference, it is not possible to compare the translations without having them examined by human subjects. This was done for the 73 such cases (49+24) of the ZH/EN post-editing system. Three annotators, working independently, considered each translation from each system (in separate batches) with respect to the reference one, and rated its meaning on a 3-point scale: 2 (good), 1 (acceptable) or 0 (wrong). To estimate the inter-annotator agreement, we computed the average absolute deviation[5] and found a value of 0.15, thus denoting very good agreement. Below, we group '2' and '1' answers into one category, called 'acceptable', and compare them to '0' answers, i.e. wrong translations.

	Post-editing = Baseline (49)		Post-editing ≠ Baseline (24)			
	ref = Pron. (40)	ref = Non-pron. (9)	ref = Pron. (20)		ref = Non-pron. (4)	
			Baseline	Post-editing	Baseline	Post-editing
Correct	36	5	9	19	1	4
Failed	4	4	11	1	3	0

Table 4.5 – Subjective evaluation on undecided cases

Table 4.5 represents the subjective evaluation on the cases where baseline and post-edited translations differ from the reference. The human evaluation shows that the systems' translations are correct in 36 out of 40 cases. This large number shows that the quality of the systems is actually higher than what can be inferred from Table 4.4 only. Conversely, in the 9 cases when the reference translation of Y is not a pronoun, only about half of the translations are correct.

Moreover, Table 4.5 represents the subjective evaluation on the cases when baseline and post-edited translations differ from the reference *and* among themselves (24 cases) as well. In such situations, it is legitimate to ask which of the two systems is better. Overall, 10 baseline translations are correct and 14 are wrong, whereas 23 post-edited translations are correct (or at least acceptable) and only one is wrong. The post-edited system thus clearly outperforms the baseline in this case. Similarly to the observation above, we note that among the 24 cases considered here, almost all (20) involve a reference translation of Y by a pronoun. In these cases, the baseline system translates only about half of them with a correct noun, while the post-edited system translates correctly 19 out of 20.

Example 2 in Figure 4.3 is one of the 19 correct cases. Although the translation of our system, 'symphony', cannot be evaluated automatically as the reference translation uses the demonstrative 'that', we can see that it is much more accurate compared to the baseline translation, 'tune'.

Example 3 in Figure 4.3 is one of the 4 cases where the reference translation is a full noun phrase and the translation generated by our system differs from the reference, but is evaluated as correct by a human. In this example, our system's translation, 'bike', is not identical to

[5] Average of $\frac{1}{3}\sum_{i=1}^{3}|\text{score}_i - \text{mean}|$ over all ratings .

‘bicycle’ from the reference, but is a synonym of it, whereas the baseline makes a mistake by using ‘car’.

	Post-editing≠ Baseline ∧ Post-editing ≠ Reference	
	Baseline	Post-editing
Average	1.01	1.38
Fluctuation	0.14	0.15

Table 4.6 – Mean and fluctuation of the human evaluation scores.

Finally, we also compute the mean and fluctuation values of the human scores, as shown on Table 4.6. Three annotators gave a score to each analyzed sentence, and we first calculate the mean value of each sentence using these three scores, then sum the mean values of all sentences and compute the average as $\frac{\sum_{i=1}^{34} mean_i}{34}$ for both baseline and our post-editing method (the number of candidate sentences is 34). In parallel, we compute the fluctuation value of the two systems by computing the variance (or consistency) of the scores compared with the mean: $\sum_{i=1}^{34} \frac{1}{3} \sum_{ij=1}^{3} |score_j - mean_i|$. These results indicate an acceptable agreement among annotators.

5 Consistent Translation of Repeated Nouns

Although leveraging the translations of compound pairs as described in Chapter 4 shows slight but constant improvements of the translation task, the necessity of a XY/Y configuration is a serious limitation of the method. In this chapter, we extend our method from Chapter 4 by focusing on the translations of all repeated nouns, which require consistent translations especially in case of co-reference. However, consistency should not be always enforced, in order to avoid the "trouble with MT consistency" [Carpuat and Simard, 2012] which may actually increase the number of translation errors.

More precisely, we generalize our approach in this chapter by checking all repeated nouns appearing in the source-side of parallel texts. We aim to improve the translation of these nouns by designing a classifier which predicts, for every such pair of nouns, whether they should be translated consistently by the same noun or not. If yes, the classifier also predicts which of the two candidate translations generated by a baseline MT system should replace the other one.

The consistency classifier is trained on a dataset extract from parallel texts, which are used to determine the translations of repeated nouns. We present the detailed data preparation used for both our classifier and the machine translation system in Section 5.2. To build the classifier, we present the proposed syntactic and semantic features in Section 5.3.

In Section 5.4, we show that integrating the decision from our consistency classifier with SMT improves Chinese-to-English and German-to-English translations. Moreover, syntactic features appear to be more useful than semantic ones. The case of more than two consecutive occurrences of the same noun will also be examined, in Section 5.4.3. Finally, we show that a combined re-ranking and post-editing approach is the most effective one, filling about 50% of the gap in BLEU scores between the baseline MT and the use of an oracle classifier.

5.1 Introduction

The repetition of a noun in a text may be due to co-reference, i.e. repeated mentions of the same entity, or to mentions of two entities of the same type. But in other cases, two occurrences of the same noun may simply convey different meanings. The translation of repeated nouns depends, among other things, on the conveyed meanings: in case of co-reference or identical senses, they should likely be translated with the same word, while otherwise they should be translated with different words, if the target language distinguishes the two meanings. State-of-the-art machine translation systems do not address this challenge systematically, and translate two occurrences of the same noun independently, thus potentially introducing unwanted variations in translation.

We exemplify the consistency issue in Figure 5.1 for Chinese-to-English and German-to-English translations, with examples of inconsistent translations of repeated source noun by a baseline SMT system, as opposed to consistent translations in the reference. In Example 1, the system's translation of the second occurrence of *politik* is mistaken and should be replaced by the first one (*policy*, not *politics*). In Example 2, although the first translation differs from the reference, it could be acceptable as a legitimate variation, although the second one (*identity documents*) is more idiomatic and more frequent. Of course, in addition to these two examples, there are other configurations involved in a consistency relation across source, candidate and reference translations, and they will be discussed in Section 5.2 when designing the training and test data for our problem.

Example 1
Source: nach einführung dieser **politik** [...] die **politik** auf dem gebiet der informationstechnik [...]
Reference: once the **policy** is implemented [...] the information technology **policy** [...]
MT: after introduction of **policy** [...] the **politics** in the area of information technology [...]

Example 2
Source: 欺诈性旅行或身份**证件**系指有下列情形之一的任何旅行或身份**证件**
Reference: Fraudulent travel or identity **document**; shall mean any travel or identity **document**
MT: 欺诈性 travel or identity **papers**. 系指 have under one condition; any travel, or identity **document**

Figure 5.1 – Inconsistent translations of repeated nouns from German (Example 1) and Chinese (Example 2) into English. While in both examples one translated noun is different from the reference, only Example 1 is truly mistaken: the second occurrence of the noun should be replaced with the first one.

Our method flexibly enforces noun consistency in discourse to improve noun phrase translation. We first detect two neighboring occurrences of the same noun in the source text, i.e. closer than a fixed distance, and which satisfy some basic conditions. Then, we consider their baseline translations by a phrase-based statistical MT system, which are identified from word-level alignments. If the two baseline translations of the repeated noun differ, then our classifier uses the source and target nouns and a large set of features (presented in Section 5.3) to decide whether one of the translations should be edited, and how. This decision will serve to post-edit and/or re-rank the baseline MT's output (Section 5.3.4). We deliberately limit the decisions of the classifier to these three options, and do not use, for instance, other candidate translations from the n-best list. Nor do we address the case when the two baseline translations are identical, but should not be so, i.e. when consistency should be decreased rather than increased. To design the classifier, we train machine-learning classifiers over examples that are extracted from parallel data and from a baseline MT system, as described in Section 5.2.2. A separate subset of unseen examples is used to test classifiers, first intrinsically and then in combination with MT.

5.2 Datasets for Studying Noun Consistency in MT

5.2.1 Corpora and Pre-processing

Our data comes from the WIT^3 Corpus[1] [Cettolo et al., 2012], a collection of transcripts of TED talks, and the UN Corpora[2], a collection of documents from the United Nations. The experiments are on Chinese-to-English and German-to-English.

We first build a phrase-based SMT system for each language pair with Moses [Koehn et al., 2007], with its default settings. Both MT systems are trained on the WIT^3 data, and are used to generate candidate translations of the UN Corpora. Then, we train classifiers on noun pairs extracted from the UN Corpora, using semantic and syntactic features extracted from both source and target sides. The test sets also come from the UN Corpora, with the same features on the source side. Table 5.1 presents statistics about the data.

WIT^3 Data	MT training		MT tuning		Language modeling	
	Sentences	Words	Sentences	Words	Sentences	Words
DE-EN	193,152	3.6M	2,052	40K	217K	4.4M
ZH-EN	185,443	3.4M	2,457	54K	4.8M	800M

UN Data	Classifier training			Classifier testing		
	Sentences	Words	Nouns	Sentences	Words	Noun
DE-EN	150K	4.5M	11,289	7,771	225K	695
ZH-EN	10K	368K	3,301	3,000	121K	647

Table 5.1 – WIT^3 data for building the SMT systems and UN data to train/test the classifiers.

5.2.2 Extraction of Training/Testing Instances

At this stage, the goal is to automatically extract as training data the pairs of repeated nouns in the source texts, noted $N \dots N$, which are translated differently by the SMT baseline, noted $T_1 \dots T_2$, with $T_1 \neq T_2$. Indeed, when the translations are identical, we have no element in the 1-best translation to post-edit them, therefore we do not consider such pairs. We examine the reference translations of T_1 and T_2, noted RT_1 and RT_2, from which we derive the answer we expect from the classifiers (as specified below), and which will be used for supervised learning.

[1] http://wit3.fbk.eu

[2] http://www.uncorpora.org

We obtain the T_i and RT_i values using word-alignment with GIZA++.

Prior to the identification of repeated nouns in the source text, we tokenize the texts and identify parts-of-speech (POS) using the Stanford NLP tools[3]. In particular, as Chinese texts are not word-segmented, we first perform this operation and then identify multi-character nouns. We then consider each noun in turn, and look for a second occurrence of the same noun in what follows, limiting the search to the same sentence for Chinese, and to the same and next three sentences for German. The difference in the distance settings is based on observations of the Chinese vs. German datasets: average length of sentences, average distance of repeated nouns, and sentence segmentation issues.

Once the pairs of repeated nouns have been identified, we check the SMT translations of each pair, and if the two translations are different, we include the pair in our dataset. For instance, in Figure 5.1, the noun '证件' appears twice in the sentence, and the baseline translations of the two occurrences are *papers* and *document*; therefore, this pair is included in our dataset. We extracted from the UN Corpora 3,301 pairs for training and 647 pairs for testing on ZH/EN, and 11,289 pairs for training and 695 pairs for testing on DE/EN. We obtained a smaller amount of noun pairs for ZH/EN than DE/EN because the latter dataset is more than 10 times larger than the former. We kept similar test set sizes to enable comparison.

The word-aligned reference translations are used to set the ground-truth class (or decision) for training the classifiers, as follows. With the notations above (baseline translations of N noted T_1 and T_2, with $T_1 \neq T_2$):

- If the reference translations are different ($RT_1 \neq RT_2$), then we label the pair as 'none', i.e. none of T_1 and T_2 should be post-edited and changed into the other, because this would not help to reach the reference translation anyway (recall that the only possible actions knowing the SMT baseline are replacing T_1 by T_2 or vice-versa).

- If the reference translations are the same ($RT_1 = RT_2$), then we examine this word, noted RT. If this word is equal to one of the baseline translations ($T_1 = RT$ or $T_2 = RT$), then this value should be given to other baseline (e.g., if $T_1 = RT \neq T_2$, then $T_2 := T_1$). For classification, we simply label these examples with the index of the word that must be used, 1 or 2.

- However, if the reference differs from both baseline translations, then the label is again 'none', because we cannot infer which of them is a better translation.

After labeling all the pairs, we extract the features in an attribute/value format to be used for machine learning.

[3] http://nlp.stanford.edu/software

5.3 Classifiers for Translation Consistency

5.3.1 Role and Nature of the Classifiers

We describe here the machine learning classifiers that are trained to predict one of the three classes – ‘1’, ‘2’ or ‘none’ – for each pair of identical source nouns with different baseline SMT translations. The sense of the predicted classes is the following: ‘1’ means that T_1 should replace T_2, ‘2’ means the opposite, and ‘none’ means translations should be left unchanged. For instance, if Example 2 in Figure 5.1 was classified as ‘2’, we would replace the translation of the first occurrence (*papers*) with the second one (*document*).

As with many other classification tasks in NLP, in this section we compare Random Forests, Decision Trees, Support Vector Machines and Maximum Entropy algorithms. We briefly define and discuss the drawbacks of these algorithms for the task under study, and draw some initial arguments for using such algorithms to improve baseline translations.

Decision Trees and Random Forests: Decision Trees (such as the C4.5 algorithm proposed by Quinlan [1993]), or an ensemble of them called a Random Forest [Breiman, 2001], have the advantage that they can easily be visualized to see which features most contribute to solve the classification task. Most often, the decisions are binary only, as they are based on a yes/no decision for a specific feature without considering all features at that decision point.

Support Vector Machines: SVMs have been used for a large range of machine learning problems and perform well because they can linearly (in the feature space) separate non-linearly separable data, thanks to the use of kernels that project the data onto an implicit, higher dimensional space. In our configurations, the maximum entropy algorithm however outperformed the SVM-based classifiers.

Maximum Entropy: MaxEnt models are discriminative and based on conditional probabilities that can be calculated from the class distribution present in the data. Their name comes from the fact that one would like the probability distributions to be as uniform as possible, by introducing only the constraints or features that help to reduce the entropy to the level that resembles the actual class distribution in the data. The main advantage of MaxEnt models is that they can learn the most useful feature associations through feature weighting and inter-dependence analysis. In addition, the output of MaxEnt models is easily interpretable, as features and classes are assigned a probability value that indicates the confidence of the classifier in its decision.

We use the WEKA environment[4] to train and test various learning algorithms: SVMs [Cortes and Vapnik, 1995], C4.5 Decision Trees (noted J48 in Weka) [Quinlan, 1993], and Random Forests [Breiman, 2001]. We use 10-fold cross-validation on the training set, and then test

[4] http://www.cs.waikato.ac.nz/ml/weka/

them once on the test set, and later on in combination with MT. For performance reasons, we used the Maximum Entropy classifier [Manning and Klein, 2003] from Stanford[5] instead of WEKA's Logistic Regression.

The hyper-parameters of the above classifiers are set as shown in Table 5.2, mostly following the default settings from WEKA, and optimizing others on the cross-validation sets (not the unseen test sets). For SVMs, the round-off error is $\epsilon = 10^{-12}$. For Decision Trees, we set the minimal number of instances per leaf ('minNumObj') at 2 and the confidence factor used for pruning to 0.25. For Random Forests, we defined the number of trees to be generated ('numTree') as 100 and set their maximal depth ('maxDepth') as unlimited. Finally, we set the MaxEnt smoothing (σ) to 1.0, and the tolerance used for convergence in parameter optimization to 10^{-5}.

Methods	**Parameter Setting**
SVM	$\epsilon = 10^{-12}$
Decision Tree	minNumObj = 2 confidence factor = 0.25

Random Forests	numTree = 100
	maxDepth = ∞

Maximum Entropy	$\sigma = 1.0$ tolerance = 10^{-5}

Table 5.2 – Parameter settings for each learning method.

5.3.2 Syntactic Features

We defined 19 syntactic features, mainly with the assumption that out of a pair of repeated source nouns $N \ldots N$, the occurrence which is embedded in a more complex parse tree, i.e. has more information syntactically bound to it, is more "determined" and has a higher probability of been translated correctly by the baseline MT system, since this information can help the system to disambiguate it. The results tend to confirm this assumption.

Source: 赞扬 联合国 人权 事务 高级 专员 办事处 高度 优先 从事 有关 国家 机构 的 工作 ， [...] ， 鼓励 高级 专员 确保 作出 适当 安排 和 提供 预算 资源
Reference: commends the high priority given by the office of the united nations high **commissioner** for human rights to work on national institutions , [...] , encourages the high **commissioner** to ensure that appropriate arrangements are made and budgetary resources provided
MT: praise the human rights high **commissioner** was the high priority to offices in the country , [...] , to encourage senior **specialists** to make sure that make appropriate and provide budget resources

Figure 5.2 – Chinese text example used for syntactic feature analysis.

The syntactic features are listed in Table 5.3, with an explicit description of each feature and

[5] http://nlp.stanford.edu/software/classifier.shtml

Features	**Values**
Source noun (Chinese)	专员
Distance in sentences between the two source occurrences	0
Translation of the first occurrence (labelled NN)	commissioner
Translation of the second occurrence (labelled NN)	specialists
Number of sibling nodes of the 1st occurrence	4
Number of sibling nodes of the 2nd occurrence	2
Sign of the difference between the above (+1, 0, -1)	1
Number of words of the 1st occurrence and its siblings	2
Number of words of the 2nd occurrence and its siblings	1
Sign of the difference between the above (+1, 0, -1)	1
Number of nodes in the first NP ancestor of 1st occurrence	16
Number of nodes in the first NP ancestor of 2nd occurrence	7
Sign of the difference between the above (+1, 0, -1)	1
Number of words in the first NP ancestor of 1st occurrence	6
Number of words in the first NP ancestor of 2nd occurrence	2
Sign of the difference between the above (+1, 0, -1)	1
Distance between the first NP ancestor and the 1st occurrence	3
Distance between the first NP ancestor and the 2nd occurrence	3
Sign of the difference between the above (+1, 0, -1)	0
Class (1, 2, 0)	1

Table 5.3 – Definition of syntactic features and illustration of their values on the sample Chinese text in Figure 5.2.

its value on a Chinese text in Figure 5.2. In the last line of the feature list in Table 5.3 we show the ground-truth class of this example.

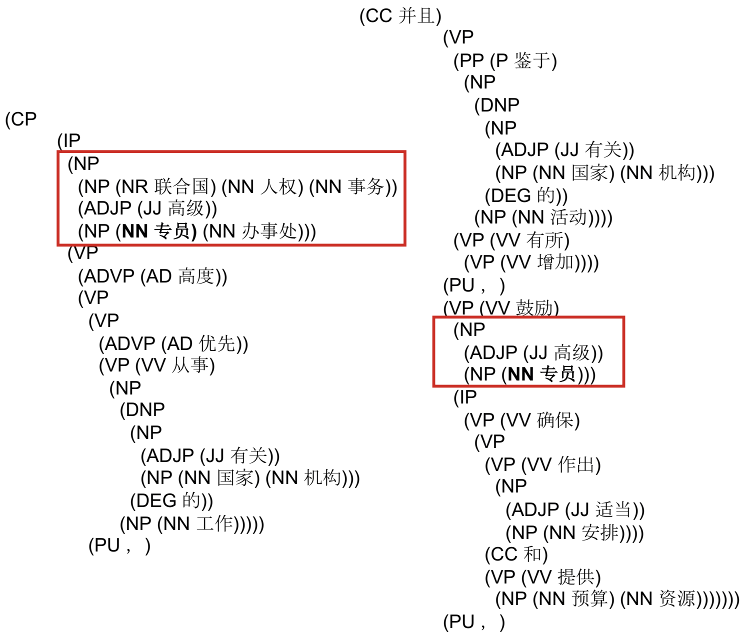
The sentences are parsed using the Stanford parser[6], and the values of the features are obtained from the parse trees, using the sizes (in nodes or words) of the siblings and ancestor sub-trees for each analyzed noun. In the sample parse trees shown in Figure 5.3, the first NP ancestor is marked with a red rectangle, and the values of the features are computed using it.

We can distinguish four subsets of features. The first subset includes lexical and positional features: the original noun, automatic baseline translations of both occurrences from the baseline MT system, and the distance between the sentences that contain the two nouns. The second subset includes features that capture the size of the siblings in the parse trees of each of the two nouns. The third subset includes the size of the sub-tree for the latest noun phrase ancestor for each analyzed noun. The last subset includes the information of the depth distances to the next noun phrase ancestor.

5.3.3 Semantic Features

The semantic features, to be used independently or in combination with the syntactic ones, are divided into two groups: discourse vs. local context features, which differ by the amount

[6] http://nlp.stanford.edu/software/lex-parser.html



```
(CP
  (IP
   (NP
    (NP (NR 联合国) (NN 人权) (NN 事务))
    (ADJP (JJ 高级))
    (NP (NN 专员) (NN 办事处)))
   (VP
    (ADVP (AD 高度))
    (VP
     (VP
      (ADVP (AD 优先))
      (VP (VV 从事)
       (NP
        (DNP
         (NP
          (ADJP (JJ 有关))
          (NP (NN 国家) (NN 机构)))
         (DEG 的))
        (NP (NN 工作)))))
     (PU ，)
(CC 并且)
     (VP
      (PP (P 鉴于)
       (NP
        (DNP
         (NP
          (ADJP (JJ 有关))
          (NP (NN 国家) (NN 机构)))
         (DEG 的))
        (NP (NN 活动))))
      (VP (VV 有所)
       (VP (VV 增加))))
     (PU ，)
     (VP (VV 鼓励)
      (NP
       (ADJP (JJ 高级))
       (NP (NN 专员)))
      (IP
       (VP (VV 确保)
        (VP
         (VP (VV 作出)
          (NP
           (ADJP (JJ 适当))
           (NP (NN 安排))))
         (CC 和)
         (VP (VV 提供)
          (NP (NN 预算) (NN 资源)))))))
     (PU ，)
```

Figure 5.3 – Parse trees obtained on the sample Chinese text. The red boxes in the parse trees show the first NP ancestors of the examined nouns.

of context they take into account. On the one hand, local context features represent the immediate context of each noun in the pair and their translations, i.e. three words to their left and three words to their right in both source and MT output, always within the same sentence.

On the other hand, discourse features capture those cases where the inconsistent translations of a noun might be due to a disambiguation problem of the source noun, and semantic similarity can be leveraged to decide which of the two translations best matches the context. To compute the discourse features, we use the word2vec word vector representations generated from a large corpus [Mikolov et al., 2013a], which have been successfully used in the recent past to compute similarity between words [Schnabel et al., 2015]. Specifically, we employ the model trained on the English Google News corpus [7] with about 100 billion words.

In the mathematical definition, the cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$\vec{a} \times \vec{b} = ||\vec{a}||||\vec{b}|| \cos\theta$$

and given two vectors of attributes, A and B, the cosine similarity is represented using a dot

[7] https://code.google.com/p/word2vec/

product and magnitude as:

$$similarity = \cos(\theta) = \frac{\vec{A} \times \vec{B}}{||\vec{A}||||\vec{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_i^n A_i^2}\sqrt{\sum_i^n B_i^2}}$$

In the Example 1 from Figure 5.1, the German word *Politik* is translated into the English words *policy* and then *politics*. The semantic similarity between the word *politics* and its context is lower than the similarity between *policy* and its context, which we consider to be an indication that the first occurrence, namely *policy*, has better chances to be the correct translation – which is actually the case in this example.

5.3.4 Integration with the MT System

The classifier outputs a post-editing decision for each pair of repeated nouns: replace T_1 with T_2, replace T_2 with T_1, or do nothing. This decision can be directly executed, or it can be combined in a more nuanced fashion with the MT system. Therefore, to modify translations using this decision, we propose and test three approaches for using it in an MT system:

Post-editing: directly edit the translations T_1 or T_2 depending on the classifier's decision.

Re-ranking: search among the translation hypotheses provided by the SMT system (in practice, the first 10,000 ones) for those where T_1 and T_2 are translated as predicted by the classifier, and select the highest ranking one as the new translation. If none is found, the baseline 1-best hypothesis is kept.

Re-ranking + Post-editing: after applying re-ranking, if no hypothesis conforms to the prediction of the classifier, instead of keeping the baseline translation we post-edit it as in the first approach.

5.4 Results and Analysis

We evaluate our proposal in two ways. First, we measure the classification accuracy in terms of *accuracy* and *kappa* (κ) agreement [Cohen, 1960] with the correct class, either in 10-fold cross-validation experiments, or on the test set. Second, we compare the updated translations with the reference, to check if we obtain a result that is closer to it, using the popular BLEU measure [Papineni et al., 2002].

We first present the results of the classification task, i.e. the prediction of the correct translation variant (1^{st} / 2^{nd} / None), for Chinese-English and German-English translation respectively in Tables 5.4 and 5.5, with 10-fold cross-validation on the training sets. Then, we present the scores on the test sets for both the classification task and its combination with MT.

Table 5.6 and 5.7 show the results on ZH/EN, according to the features that are used: syntactic or semantic. We then merge the feature sets and show the result in Table 5.8 using all features. We present the results for DE/EN in a similar way, in Tables 5.9, 5.10 and 5.11.

We compare the results obtained with several ML methods: Decision Trees (J48), SVMs, Random Forests and MaxEnt, ordered in the tables by *average* increasing scores. Moreover, we compare the merits of syntactic vs. semantic features, as well as post-editing vs. re-ranking the MT output.

	Syntactic features		Semantic features		All features	
	Acc. (%)	κ	Acc. (%)	κ	Acc. (%)	κ
J48	72.1	0.48	60.2	0.00	60.2	0.00
SVM	74.5	0.54	60.2	0.00	73.9	0.51
RF	75.3	0.54	68.4	0.29	70.7	0.35
MaxEnt	76.7	0.65	69.5	0.32	**83.3**	**0.75**

Table 5.4 – Prediction of the correct translation (1st / 2nd / None) for repeated nouns in *Chinese*, in terms of accuracy (%) and kappa scores, *on the development set* with 10-fold cross-validation. Methods are sorted by average accuracy over the three feature sets. When using semantic or all features, no decision tree outperformed the majority class baseline, hence $\kappa = 0$. The best scores are in bold.

	Syntactic features		Semantic features		All features	
	Acc. (%)	κ	Acc. (%)	κ	Acc. (%)	κ
SVM	77.8	0.67	38.1	0.00	38.1	0.00
J48	77.0	0.66	64.8	0.45	79.7	0.69
RF	82.0	0.73	73.5	0.60	**84.5**	**0.77**
MaxEnt	80.8	0.71	76.8	0.65	83.4	0.75

Table 5.5 – Prediction of the correct translation (1st / 2nd / None) for repeated nouns in *German*, in terms of accuracy (%) and kappa scores, *on the development set* with 10-fold c.-v. Methods are sorted by average accuracy over the three feature sets. The best scores are in bold.

5.4.1 Best Scores of Classification and MT

The classification accuracy is above 80% when applying 10-fold cross-validation, for both language pairs, and reaches 74–78% on the test sets. As the classes are quite balanced, a random baseline would reach around 33% only. Kappa values reach 0.75 on the training sets and 0.60–0.67 on the test sets. The performances of the classifiers appear thus to be well above chance, and the high performances achieved on the unseen test sets indicate that over-fitting is unlikely.

The ordering of methods by performance is remarkably stable: Decision Trees (J48) and SVMs get the lowest scores, followed by Random Forests, and then by the MaxEnt classifier. The ordering {J48, SVM} < RF < MaxEnt is observed over both language pairs, over the three types of features, and the four datasets, with 1-2 exceptions only. Overall, the best configuration of

	Chinese: Syntactic features				
	Acc.	κ	BLEU		
			PE	RR	RR+PE
Baseline	-	-	11.07	11.07	11.07
J48	66.3	0.42	11.17	11.20	11.30
SVM	71.9	0.53	11.23	11.27	11.33
RF	71.7	0.53	11.22	11.24	11.27
MaxEnt	73.7	0.60	11.27	11.33	**11.35**
Oracle	100	1.00	11.40	11.52	11.64

Table 5.6 – Prediction of the correct translation learned by syntactic features (accuracy (%) and *kappa*) and translation quality (BLEU) for repeated nouns on the *Chinese test set*. Maximum Entropy was the best method found on the dev set compare to other methods trained by syntactic features.

	Chinese: Semantic features				
	Acc.	κ	BLEU		
			PE	RR	RR+PE
Baseline	-	-	11.07	11.07	11.07
J48	33.1	0.00	11.07	11.07	11.07
SVM	33.1	0.00	11.07	11.07	11.07
RF	55.2	0.33	11.04	11.07	11.12
MaxEnt	56.1	0.34	10.87	11.11	11.18
Oracle	100	1.00	11.40	11.52	11.64

Table 5.7 – Prediction of the correct translation learned by semantic features (accuracy (%) and *kappa*) and translation quality (BLEU) for repeated nouns on the *Chinese test set*.

our method found on the training sets is, for both languages, the MaxEnt classifier with all features.

There is a visible rank correlation between the increase in classification accuracy and the increase in BLEU score, for all languages, features, classifiers, and combination methods with MT. The best configurations found on the training sets bring the following BLEU improvements: for ZH/EN, from 11.07 to 11.36, and for DE/EN, from 17.10 to 17.67. In fact, syntactic features turn out to reach an even higher value on the test set, at 17.75. To interpret these improvements, they should be compared to the oracle BLEU scores obtained by using a “perfect” classifier, which are 11.64 for ZH/EN and 17.99 for DE/EN. Our method thus bridges 51% of the BLEU gap between baseline and oracle on ZH/EN and 64% on DE/EN – a significant improvement.

The BLEU scores of the three different integration methods for using classification for MT (Tables 5.8 and 5.11) clearly show that the combined method outperforms both post-editing and re-ranking alone, for all languages and features. Post-editing, the easiest one to implement, has little consideration for the words surrounding the nouns, while re-ranking works on MT

	Chinese: All features				
	Acc.	κ	BLEU		
			PE	RR	RR+PE
Baseline	-	-	11.07	11.07	11.07
J48	33.1	0.00	11.07	11.07	11.07
SVM	62.1	0.43	11.18	11.26	11.26
RF	54.9	0.32	11.16	11.20	11.24
MaxEnt	72.5	0.56	11.21	11.33	**11.36**
Oracle	100	1.00	11.40	11.52	11.64

Table 5.8 – Prediction of the correct translation (accuracy (%) and *kappa*) and translation quality (BLEU) for repeated nouns on the *Chinese test set.* Maximum Entropy was the best method found on the dev set compare to other methods trained by all features.

	German: Syntactic features				
	Acc.	κ	BLEU		
			PE	RR	RR+PE
Baseline	-	-	17.10	17.10	17.10
SVM	71.4	0.57	17.59	17.65	17.72
J48	70.5	0.56	17.59	17.61	17.70
RF	70.2	0.55	17.55	17.62	17.68
MaxEnt	78.3	0.67	17.63	17.66	**17.75**
Oracle	100	1.00	17.78	17.83	17.99

Table 5.9 – Prediction of the correct translation (accuracy (%) and *kappa*) and translation quality (BLEU) for repeated nouns on the *German test set.* Maximum Entropy learned by syntactic features was the best method found on the dev set.

hypotheses and thus ensures that a better global translation is found that is also consistent. However, in some cases, no hypothesis conforms to the consistency decision, and in this case post-editing the best hypothesis appears to be beneficial.

5.4.2 Feature Analysis: Syntax vs. Semantics

On the training sets, syntactic features always outperform the semantic ones when using the MaxEnt classifier, and their joint use outperforms their separate uses. For the other classifiers (not the best ones on the training sets), on ZH/EN, adding semantic features to syntactic ones decreases the performance. Indeed, semantic features (specifically the discourse ones) are intended to disambiguate nouns based on contexts, but here, manual inspection of the data showed that these are similar for T_1 and T_2, which makes prediction difficult.

Semantic features appear to be more useful in German compared to Chinese. We hypothesize that this is because translation ambiguities of Chinese nouns, i.e. cases when the same noun can be translated into English with two very different words, are less frequent and less seman-

	German: Semantic features				
	Acc.	κ	BLEU		
			PE	RR	RR+PE
Baseline	-	-	17.10	17.10	17.10
SVM	32.8	0.00	17.10	17.10	17.10
J48	48.2	0.23	17.13	17.27	17.33
RF	54.4	0.32	17.21	17.34	17.37
MaxEnt	63.5	0.49	17.39	17.47	17,49
Oracle	100	1.00	17.78	17.83	17.99

Table 5.10 – Prediction of the correct translation (accuracy (%) and *kappa*) and translation quality (BLEU) for repeated nouns on the *German test set*.

	German: All features				
	Acc.	κ	BLEU		
			PE	RR	RR+PE
Baseline	-	-	17.10	17.10	17.10
SVM	32.8	0.00	17.10	17.10	17.10
J48	69.4	0.54	17.56	17.60	17.66
RF	67.6	0.52	17.53	17.57	17.63
MaxEnt	68.7	0.53	17.58	17.59	**17.67**
Oracle	100	1.00	17.78	17.83	17.99

Table 5.11 – Prediction of the correct translation (accuracy (%) and *kappa*) and translation quality (BLEU) for repeated nouns on the *German test set*. Maximum Entropy learned by all features was the best method found on the dev set.

tically divergent than in German. In other words, semantic features are less useful in Chinese because cases of strong polysemy or homonymy seem to be less frequent than in German, for the following reason.

A lexical item is polysemous if it can convey several related senses, while homonyms are identical lexical items with unrelated senses. The various senses of a polysemous noun can be rendered in translation by different (although neighboring words) or by a similarly polysemous word. However, two homonyms are generally translated by very different words, unless the homonymy has been borrowed in one language from another. Homophony is known to be frequent in Chinese, but does not concern written words: indeed, a study of Chinese nouns and verbs [Huang, 1995] brings evidence for our statement and concludes: "words in Chinese, dimorphemic words in particular, are as a rule much less polysemic than those in English." In Huang's examination of a random sample of about 14,000 English nouns, about 33% were polysemous (or homonyms), with about 3 senses per noun, while in a sample of nearly 6,000 Chinese nouns (with several morphemes) only 22% were polysemous, with fewer than 2 senses per noun. We believe that German behaves similarly to English in this respect.

These facts might also explain the results obtained when using all features, for German and

Chinese. As in Chinese semantic features are less helpful, given also the limited amount of data, combining them with syntactic ones actually decreases the performance of the syntactic ones used independently. In contrast, semantic features are more helpful on German dataset, and also improve results when considered together with the syntactic ones.

ZH/EN	
Translation of the 2nd occurrence	0.165
Translation of the 1st occurrence	0.163
Source noun	0.110
Number of words in the first NP ancestor of the 2nd occ.	0.060
Number of words in the first NP ancestor of the 1st occ.	0.050
Number of nodes in the first NP ancestor of the 2nd occ.	0.036
Number of nodes in the first NP ancestor of the 1st occ.	0.033
Sign of difference between number of words and its siblings	0.031
Distance between the first NP ancestor and the 2nd occ.	0.025
Number of words of the 1st occ. and its siblings	0.023

Table 5.12 – Top ten syntactic features ranked by information gain for Chinese-to-English language pair.

DE/EN	
Translation of the 1st occurrence	0.162
Translation of the 2nd occurrence	0.162
Source noun	0.099
Number of words in the first NP ancestor of the 2nd occ.	0.062
Number of words in the first NP ancestor of the 1st occ.	0.057
Number of nodes of the 2nd occ.	0.054
Number of sibling nodes of the 1st occ.	0.052
Number of nodes in the first NP ancestor of the 1st occ.	0.042
Number of nodes in the first NP ancestor of the 2nd occ.	0.037
Number of words of the 2nd occ. and its siblings	0.037

Table 5.13 – Top ten syntactic features ranked by information gain for German-to-English language pair.

Tables 5.12 and 5.13 show the top ten syntactic features for ZH/EN and for DE/EN, ranked by information gain computed using Weka. These features include both lexical information and properties of the parse trees. The analysis shows that lexical features are significantly more important than purely syntactic ones, for both languages, but the importance of syntactic features is not negligible at all.

5.4.3 Extension to Triples of Repeated Nouns

Finally, we consider the case of nouns that appear more than twice. Using our dataset, we identify them as noun pairs that share the same word, i.e. triples of repeated nouns, to which we limit our investigation. There are 129 triples in ZH/EN and 138 DE/EN.

In our previous experiment, we considered triples of nouns only by splitting them into two pairs and deciding each translation through our system. We block the updated translation of the nouns after each measurement.

Here we define the following method to determine the translation of such triples of nouns when their baseline translations are different across the two pairs. If T_1, T_2 and T_3 are the translation candidates, we aim to find the most consistent translation T_c as follows:

- If two of the T_i are identical, we use this value as T_c.
- If they all differ, then we compare the syntactic features of the three source occurrences, and select the one with the highest number of features with highest values, and use its value as T_c.
- Going back to our classifier, if the decision for a particular instance pair is not "none", we replace the translations of the instance pairs with T_c.

		Syntactic features		Semantic features		All feature	
		= ref	≠ ref	= ref	≠ ref	= ref	≠ ref
ZH/EN	J48	79	50	-	-	41	88
	SVM	87	42	-	-	75	54
	RF	85	44	72	57	74	55
	MaxEn	97	32	67	62	88	41
DE/EN	J48	58	80	26	112	-	-
	SVM	61	77	-	-	57	81
	RF	82	65	29	109	51	87
	MaxEn	66	72	49	89	61	77

Table 5.14 – Confusion matrix of identical cases compared with reference over all classifiers over two language pairs, for triples of nouns.

We first investigate the translation quality of triples before using our specific rules and represent the confusion matrix of our translation output compared with reference in Table 5.14. The results are from a total of 129 triples in the test set, and show that our pair-wise decision system performs resonably well on triples.

However, we also consider the cases in which the updated translations are not the same after the two separate modifications, since the triples require two decisions, for A/B and B/C. The shared word would face possibly two modifications. If the modifications are the same, then there is no need to extract them and analyze any further. Otherwise, we use the rules above to make a decision on which modification to perform.

We represent the number of correct cases (identical to reference) before and after our triple modification through the method above, and also BLEU scores with this output in Table 5.15.

		Syntactic features			Semantic features			All features		
			Cases	BLEU		Cases	BLEU		Cases	BLEU
ZH/EN	J48	37	13 ↑	0.01 ↑	-	-	-	-	-	-
	SVM	26	6 ↑	0.01 ↑	-	-	-	38	4 ↑	-
	RF	28	2 ↓	-	19	2 ↓	0.01 ↓	13	5 ↓	-
	MaxEn	27	3 ↑	0.01 ↑	47	20 ↑	0.01 ↓	35	9 ↑	0.01 ↑
DE/EN	J48	52	34 ↑	0.01 ↑	70	39 ↑	0.01 ↑	57	29 ↑	0.04 ↑
	SVM	58	35 ↑	0.02 ↑	-	-	-	-	-	-
	RF	65	29 ↑	-	66	37 ↑	-	64	37 ↑	0.03 ↑
	MaxEn	50	39 ↑	0.03 ↑	78	43 ↑	0.02 ↑	61	32 ↑	0.06 ↑

Table 5.15 – BLEU results when considering rules for triples of nouns: "-" indicates no occurrences or no changes compare to pairwse results.

We experiment with the three feature types and the four classifiers, i.e. 12 cases per language. As shown in Table 5.15, on ZH/EN, a small increase of BLEU is observed in 5 cases (0.01), a decrease in two cases (0.02), and no variation in 5 cases. On DE/EN, half of the cases show a small improvement (up to 0.03) and the rest stay the same. The method appears to work better on DE/EN, possibly because the initial accuracy on pairs is lower, but all improvements are very small. The main conclusion from experimenting with triples, and considering also longer lexical chains of consistent nouns, is that the pairwise method should be replaced by a different type of consistency predictor, which remains to be found.

5.5 Conclusion of Part I

In first part of the thesis, we started by presenting a method that enforces the consistent translation of co-references to a compound, when the co-reference matches the head noun of the compound. Experimental results showed that baseline SMT systems often translate co-references to compounds consistently for DE/FR, but much less so for ZH/EN. For a significant number of cases in which the noun phrase Y had multiple meanings, our system reduced the frequency of mistranslations in comparison to the baseline, and improved noun phrase translation.

We generalized our approach to noun phrases that are not compounds, and presented a method for enforcing consistent translations of repeated nouns, by using a machine learning approach with syntactic and semantic features to decide when consistency should be enforced. We experimented with Chinese-English and German-English data. To build our datasets, we detected source-side nouns which appeared twice within a fixed distance and were translated differently by baseline MT. Syntactic features were defined based on the complexity of the parse trees containing the nouns, capturing mainly which of the two occurrences of a noun is more syntactically bound, while semantic features focused on the similarity between each translated noun and its context. The trained classifiers have shown that they can predict consistent translations above chance, and that, when combined to MT, bridge 50–60% of the gap between the baseline and an oracle classifier.

In the second part of the thesis, instead of correcting the translation with lexical consistency knowledge, we will rather design machine translation systems that automatic select the correct translation during the generation process by considering contextual information.

Word Sense Disambiguation for Machine Translation

Part II

6 Previous Work on Sense Selection for MT

In the first part of the thesis, we attempted to correct the translations of ambiguous words by using lexical consistency. Lexical consistency helped us select the proper translation when a word was repeated, but cannot address the case of non repeated words. In the second part, we will try to find the sense which is signaled by a word in a specific context, and attempt to correct its translation based on this information, instead of correcting it based on the translation of repeated occurrences. This proposal to integrate word sense disambiguation (WSD) systems into MT is also an implicit proposal for using a larger context in MT.

6.1 Sense Integration for SMT

Many words, and particularly nouns, may have multiple meanings depending on their context of use. Word sense disambiguation (WSD) aims to identify which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple possible meanings. Lesk [1986] first investigated a WSD system making use of external resources, namely the definitions of word senses from a dictionary. Lesk's algorithm disambiguated words in their context, by comparing the definition (or gloss) of each sense of a word to the other words in the context. The word was assigned the sense whose definition shared the largest number of words in common with the glosses of the other words. Later, Banerjee and Pedersen [2002] improved Lesk's approach to take advantage of the numerous semantic relations included in WordNet [Fellbaum, 1998], and this information was shown to improve the accuracy of WSD. Many other approaches to WSD have been proposed and several toolkits for WSD with the senses of WordNet exist for English [Navigli, 2009].

Since word sense disambiguation aims to identify the sense of a word appearing in a given context [Agirre and Edmonds, 2007], it should be useful, in particular, for lexical choice in MT. In initial investigations, Carpuat and Wu [2005] obtained negative results when they tried to combine a general Chinese WSD system with a word-based SMT system. Later on, they found however a positive effect when they added WSD scores as an additional context-dependent feature function to a phrase-based SMT system [Carpuat and Wu, 2007]. Another study [Vickrey et al., 2005] reformulated the task of WSD for SMT as predicting possible target translations rather than senses of ambiguous source words, and showed that such a modified WSD improved SMT. Subsequent studies which adopted this formulation [Cabezas and Resnik, 2005, Chan et al., 2007, Carpuat and Wu, 2007] successfully integrated WSD with hierarchical or phrase-based SMT. These systems yielded marginally better translations compared to SMT baselines (0.15–0.30 BLEU).

Although the WSD reformulation proved helpful for SMT, it did not determine whether actual source-side senses are helpful or not for end-to-end SMT. Xiong and Zhang [2014] attempted to answer this question by performing word sense induction for large scale data. In particular, they proposed a topic model that automatically learned sense clusters for words in the source language. In this way, on the one hand, they avoided using a pre-specified inventory of word senses as traditional WSD does, but on the other hand, they created the risk of discovering sense clusters which do not correspond to the common senses of words needed for MT. Hence, this study left open an important question, namely whether WSD based on semantic resources such as WordNet can be successfully integrated with SMT.

Neale et al. [2016] attempted such an integration, by using a WSD system based on a sense graph from WordNet. The authors used it to specify the senses of the source words and integrate them as contextual features with a MaxEnt-based translation model for English-Portuguese MT. Similarity, Su et al. [2015] built a large weighted graph model of both source and target word dependencies and integrated them as features to a SMT model. However, apart from the sense graph, WordNet provides also textual information such as sense definitions and examples, which should be useful for WSD, but was not used in the above studies. Here, we will use this information to perform sense induction on source-side monolingual data.

6.2 Word Sense Specification for NMT

Recently, researchers applied neural networks to various domains, with frequent improvements over the state-of-the-art performance. In natural language processing, neural networks have enabled low-dimensional representations of words through *word embeddings*. However, these models treat word sense ambiguity implicitly since the different senses of a word type share the same embeddings. Several approaches, however, attempted to distinguish the representations of the different senses of a given word type.

Hill et al. [2017] demonstrated that embeddings created by monolingual models tend to model pragmatic relations between words (e.g. *teacher* being related to *student*) while those created from NMT models are more oriented towards conceptual similarity (*teacher* is related to *professor*). It would thus be more promising to induce senses for NMT based on language pairs instead of labeling them universally.

Neural MT has recently emerged as the new state of the art [Sutskever et al., 2014, Bahdanau et al., 2015, Vaswani et al., 2017]. Instead of working directly at the discrete symbol level as

SMT, it projects and manipulates the source sequence of discrete symbols in a continuous vector space. However, NMT generates only one embedding for each word type, regardless of its different senses, as analyzed for instance by Hill et al. [2017].

Several studies proposed efficient non-parametric models for monolingual word sense representation [Neelakantan et al., 2014, Li and Jurafsky, 2015, Bartunov et al., 2016, Mikolov et al., 2013a,b, Bengio et al., 2003], although they also generate a single word embedding per word type, without discriminating the differents senses of the word type. Recently, there is a rising interest in discovering embeddings for each sense of the word type [Liu et al., 2015, Chen et al., 2014]. Huang et al. [2012] tried to learn multi-sense embedding by pre-clustering the contexts of a word type into discriminated senses and using the clustering information to learn sense embeddings. However, as pre-clustering, these methods cannot jointly learn sense-discriminated embedding and clustering.

Neelakantan et al. [2014] proposed an efficient non-parametric model for word sense representation, while Li and Jurafsky [2015] proposed to use the Chinese Restaurant Process for unsupervised WSD. Later, Bartunov et al. [2016] integrated the cluster results as a weighted parameter to skip-gram for the final embedding presentation. Recently, Liu et al. [2017] focused on the particular case of homographs, i.e. word types with two unrelated senses. All these studies leave open the question whether sense representations can help neural MT by reducing word ambiguity.

A few recent studies integrated automatic sense assignment with neural MT based on the above approaches for sense-specific embeddings. Rios et al. [2017] integrated the sense knowledge generated by the SenseGram toolkit as additional features to NMT. Choi et al. [2017] reduced word ambiguity by merging context information obtained from PCA projection on both sides of parallel data for an NMT encoder and decoder.

7 Word Sense Disambiguation

In the first part of the thesis, we attempted to correct translation through lexical consistency, that is, by leveraging the translation of word pairs (either compound pairs or repeated noun pairs). In the second part, we extend our work by focusing on individual words and design a sense-aware MT that addresses word ambiguity during translation.

This chapter is dedicated to an adaptive WSD system which is later used for MT, and uses for this task a larger context than is accessible to state-of-the-art MT. Section 7.1 describes our WSD system, which performs context-dependent clustering of word occurrences and is initialized with knowledge from WordNet, in the form of vector representations of definitions or examples for each sense. Our results, presented in Section 7.2, show that our WSD system is competitive on the SemEval 2010 WSD task. We will present the integration with both SMT and NMT respectively in the following Chapters 8 and 9. With respect to previous work on WSD, we innovate on the following points:

- we present three sense clustering methods with explicit knowledge (WordNet definitions or examples) to disambiguate polysemous nouns and verbs;
- we represent each token by its context vector, obtained from word2vec word vectors in a large window surrounding the token;
- we adapt the possible number of senses per word to the ones observed in the training data rather than constraining them by the full list of senses from WordNet.

7.1 Adaptive Sense Clustering for MT

7.1.1 Overview

In this section, we present the three unsupervised or weakly supervised WSD methods used in our experiments, which all perform some form of clustering of occurrences according to their senses, as represented in Figure 7.1. All methods consider the source words that have

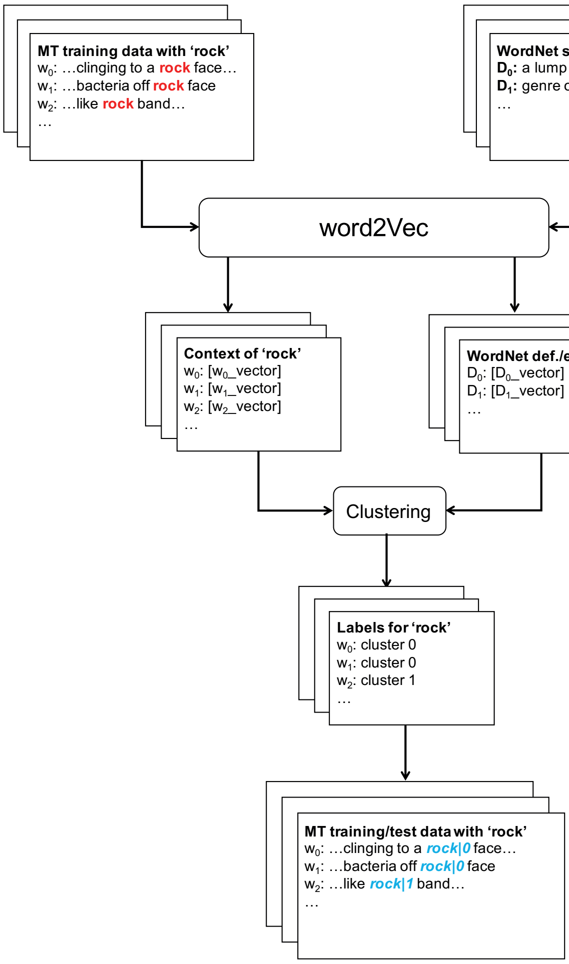
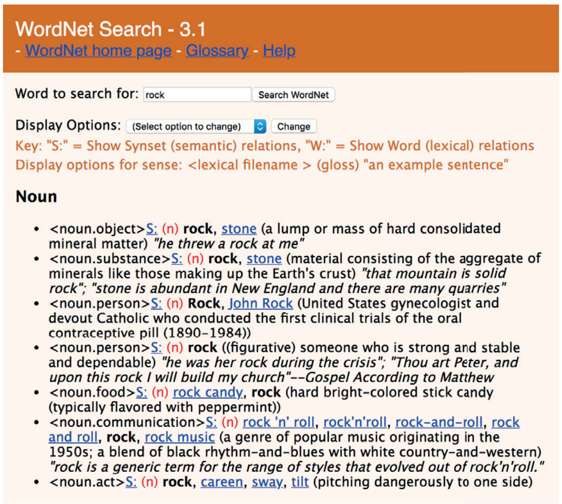

MT training data with 'rock'
w_0: ...clinging to a rock face...
w_1: ...bacteria off rock face
w_2: ...like rock band...
...
WordNet senses for 'rock'
D_0: a lump or mass of hard...
D_1: genre of popular music...
...
word2Vec
Context of 'rock'
w_0: [w_0_vector]
w_1: [w_1_vector]
w_2: [w_2_vector]
...
WordNet def./ex.
D_0: [D_0_vector]
D_1: [D_1_vector]
...
Clustering
Labels for 'rock'
w_0: cluster 0
w_1: cluster 0
w_2: cluster 1
...
MT training/test data with 'rock'
w_0: ...clinging to a rock|0 face...
w_1: ...bacteria off rock|0 face
w_2: ...like rock|1 band...
...

Figure 7.1 – Adaptive WSD for MT: vectors from WordNet definitions (or examples) are clustered with context vectors of each occurrence (here of *'rock'*), resulting in sense labels used as factors for MT.

more than one sense (synset) in WordNet, and first extract from WordNet the definition of each sense and, if available, the example provided by WordNet.

Figure 7.2 represents the sense information of a sample word, 'rock', provided by WordNet, where seven possible senses for the noun POS are listed. For each sense, WordNet provides a related definition, and also possibly an example of use, plus a lexical atribute. For instance, the first sense of 'rock', defined as "a lump or mass of hard consolidated mineral matter", has the lexical attribute of "<object>". Here, "he threw a rock at me" is given as an example. In our experiment, we filter out the senses with the lexical attribute equal to "<person>" since we do not wish to specify the related sense for proper names.

For the definitions and possibly for the examples of the senses remaining after filtering, we build word embeddings using word2vec. For each occurrence of these words in the training data, we also build vectors for their context (i.e. neighboring words) using the same model. All



WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: rock Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: <lexical filename > (gloss) "an example sentence"

Noun

- <noun.object>S: (n) **rock**, stone (a lump or mass of hard consolidated mineral matter) *"he threw a rock at me"*
- <noun.substance>S: (n) **rock**, stone (material consisting of the aggregate of minerals like those making up the Earth's crust) *"that mountain is solid rock"; "stone is abundant in New England and there are many quarries"*
- <noun.person>S: (n) **Rock**, John Rock (United States gynecologist and devout Catholic who conducted the first clinical trials of the oral contraceptive pill (1890-1984))
- <noun.person>S: (n) **rock** ((figurative) someone who is strong and stable and dependable) *"he was her rock during the crisis"; "Thou art Peter, and upon this rock I will build my church"--Gospel According to Matthew*
- <noun.food>S: (n) rock candy, **rock** (hard bright-colored stick candy (typically flavored with peppermint))
- <noun.communication>S: (n) rock 'n' roll, rock'n'roll, rock-and-roll, rock and roll, **rock**, rock music (a genre of popular music originating in the 1950s; a blend of black rhythm-and-blues with white country-and-western) *"rock is a generic term for the range of styles that evolved out of rock'n'roll."*
- <noun.act>S: (n) **rock**, careen, sway, tilt (pitching dangerously to one side)

Figure 7.2 – Sample sense information for 'rock' provided by WordNet, which contains seven possible senses for 'rock' as a noun.

the vectors are passed to a clustering algorithm, resulting in the labeling of each occurrence with a cluster number that will be used as a sense information for MT integration.

Our method answers several limitations of previous supervised or unsupervised WSD methods (reviewed in Chapter 6). Supervised methods require data with manually sense-annotated labels and are therefore often limited to a small number of word types: for instance, only 50 nouns and 50 verbs were targeted in SemEval 2010[1] [Manandhar et al., 2010]. On the contrary, our methods do not require labeled texts for training, and apply to all word types with multiple senses in WordNet.

Unsupervised methods often pre-define the number of possible senses for each ambiguous word before clustering the various occurrences according to the senses. If these numbers come from WordNet, the senses may be too fine-grained for the needs of translation, especially when a specific domain is targeted. In contrast, as we explain below, we initialize our WSD clustering algorithms with information from WordNet senses for each word (nouns and verbs), but then adapt the number of clusters to the observed training data for MT.

7.1.2 Definitions and Notations

For each noun or verb type W_t appearing in the training data, as identified by the Stanford POS tagger[2], we extract the senses associated to it in WordNet[3] [Fellbaum, 1998] using NLTK[4]. Specifically, we extract the set of definitions $D_t = \{d_{tj} | j = 1, \ldots, m_t\}$ and the set of examples of use $E_t = \{e_{tj} | j = 1, \ldots, n_t\}$, each of them containing multiple words. While most of the senses

[1] www.cs.york.ac.uk/semeval2010_WSI

[2] http://nlp.stanford.edu/software/

[3] http://wordnet.princeton.edu/

[4] http://www.nltk.org/howto/wordnet.html

are accompanied by a definition, only a smaller subset (about half the size) also include an example of use.

Definitions d_{tj} and examples e_{tj} are represented by vectors defined as the average of the word embeddings over all the words constituting them (except stopwords). Formally, these vectors are $\mathbf{d}_{tj} = (\sum_{w_l \in d_{tj}} \mathbf{w}_l)/m_t$ and respectively $\mathbf{e}_{tj} = (\sum_{w_l \in e'_{tj}} \mathbf{w}_l)/n_t$. While the entire definition d_{tj} is used to build the vector, we do not consider all words in the example e_{tj}, but limit the sum to an e'_{tj} contained in a window of size c centered around the considered word, to avoid noise from long examples.

All the word vectors $\mathbf{w}_l$ above are pre-trained word2vec embeddings from Google[5] [Mikolov et al., 2013b]. If dim is the dimensionality of the word vector space, then all vectors $\mathbf{w}_l$, $\mathbf{d}_{tj}$, and $\mathbf{e}_{tj}$ are in $\mathscr{R}^{dim}$. Each definition vector $\mathbf{d}_{tj}$ or example vector $\mathbf{e}_{tj}$ for a word type W_t will be considered as a center vector for each sense during the clustering procedure.

Turning now to tokens, each word occurrence w_i in a source sentence is represented by the average vector $\mathbf{u}_i$ of the words from its context, which is defined as a window of c words centered in w_i. The value c of the context size is even, since we calculate the vector $\mathbf{u}_i$ for w_i by averaging vectors from $c/2$ words before w_i and from $c/2$ words after it. We stop nevertheless at the sentence boundaries, and filter out stopwords before averaging.

We will now explain how to cluster according to their sense all vectors $\mathbf{u}_i$ for the occurrences w_i of a given word type W_t, using as initial centers either the definition or the example vectors. In this way, we adapt the three clustering algorithms to our needs for WSD in an MT context, before comparing their merits empirically in Section 8.2. The objective is to cluster all occurrences w_i of a given word type W_t, represented as word vectors $\mathbf{u}_i$, according to the similarity of their senses, as inferred from the similarity of the context vectors.

7.1.3 k-means Clustering

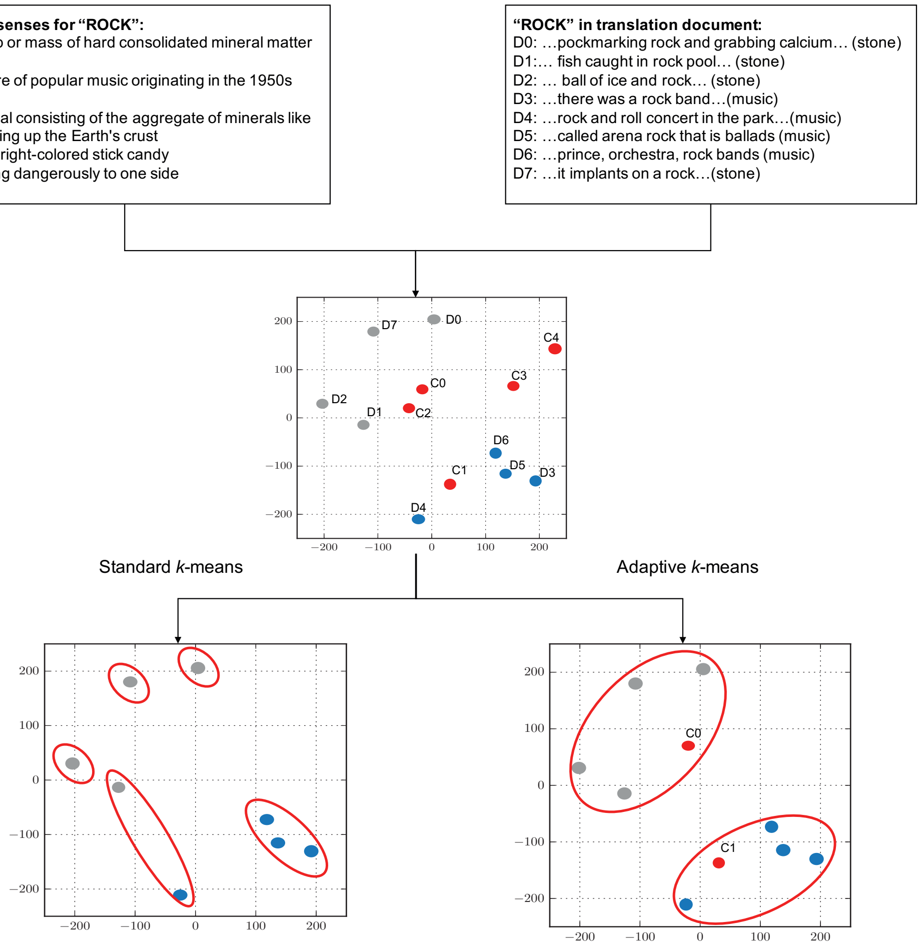
For each word type W_t, we initialize the centroids of the k-means algorithm to the vectors representing the senses from WordNet, either using their definition vectors $\mathbf{d}_{tj}$, or their example vectors $\mathbf{e}_{tj}$, hence resulting in two sets of scores that will be compared. After running the k-means algorithm, we reduce the number of clusters by merging the clusters which contain fewer than 10 tokens with the nearest larger cluster, by calculating the cosine similarity between each token vector and the centroids of the larger clusters. This re-clustering adapts the final number of clusters to the observed occurrences in the training data.

Figure 7.3 displays an example of the clustering process for the word ‘rock’ extracted from the source-side of parallel corpus. We consider all occurrences of ‘rock’ from the source-side of our corpus and compare the cluster distribution with our adaptive k-means and standard one (with center randomization). WordNet contains 5 senses for ‘rock’ as noun, so we extract

[5] code.google.com/archive/p/word2vec/

the definition for each provided sense and define the cluster k as 5 in k-means. We run the clustering procedure with both systems, adaptive and standard k-means. The adaptive k-means method discovers two dominant clusters for the senses 'stone' and 'music' by re-clustering small clusters, while the standard k-means discovers five clusters, three of which are singletons – which is expected since k was equal to the number of Wordnet senses for 'rock' (5).

We select 7 samples together with their context and manually analyze the resulting senses by projecting their position into a 2-dimensional space from the 300-dimensional vectors using T-SNE. When comparing the clusters resulting from the two systems in Figure 7.3, the gray dots annotated as 'stone' by a human are all clustered to cluster 'C0' while the gray ones annotated as 'music' are clustered into the 'C1' cluster by our adaptive system. In contrast, the standard k-means method generates clusters that are more scattered.



WordNet senses for "ROCK":
C0: a lump or mass of hard consolidated mineral matter (stone)
C1: a genre of popular music originating in the 1950s (music)
C2: material consisting of the aggregate of minerals like those making up the Earth's crust
C3: hard bright-colored stick candy
C4: pitching dangerously to one side

"ROCK" in translation document:
D0: …pockmarking rock and grabbing calcium… (stone)
D1:… fish caught in rock pool… (stone)
D2: … ball of ice and rock… (stone)
D3: …there was a rock band…(music)
D4: …rock and roll concert in the park…(music)
D5: …called arena rock that is ballads (music)
D6: …prince, orchestra, rock bands (music)
D7: …it implants on a rock…(stone)

Figure 7.3 – Clustering example of context vectors for the word 'rock' projected with t-SNE [Maaten and Hinton, 2008] using our adaptive k-means and the standard k-means method. Grey points specify the 'stone' sense, the blue points specify the 'music' sense, and the red points are the centroids according to the available definitions from WordNet. The red circles mark the clusters found by the clustering methods.

The original k-means algorithm [MacQueen, 1967] aims to partition a set of items, which are here tokens $w_1, w_2, \ldots, w_n$ of a same word type W_t, represented through their embeddings $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ where $\mathbf{u}_i \in \mathscr{R}^d$. The goal of k-means is to partition (or cluster) them into k sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares, as follows:

$$S = \arg\min_S \sum_{i=1}^{k} \sum_{\mathbf{u} \in S_i} ||\mathbf{u} - \mu_i||^2, \tag{7.1}$$

where μ_i is the centroid of each set S_i. At the first iteration, when there are no clusters yet, the algorithm selects k random points to be the centroids of the k clusters. Then, at each subsequent iteration t, k-means calculates for each candidate cluster a new point to be the centroid of the observations, defined as their average vector, as follows:

$$\mu_i^{t+1} = \frac{1}{|S_i^t|} \sum_{\mathbf{u}_j \in S_i^t} \mathbf{u}_j \tag{7.2}$$

We make the following modifications to the original k-means algorithm, starting from the implementation provided in Scikit-learn [Pedregosa et al., 2011], to make it adaptive to the word senses observed in the training data.

1. We define the initial number of clusters k_t for each ambiguous word type W_t in the data as the number of its senses in WordNet (but this number may be reduced by the final re-clustering described below at point 3). Specifically, we run two series of experiments (the results of which will be compared in Section 8.2.1): one in which each k_t is set to m_t, i.e. the number of senses that possess a definition in WordNet, and another one in which we consider only senses that are illustrated with an example, hence setting each k_t to n_t. These settings avoid fixing the number of clusters k_t arbitrarily for each ambiguous word type.

2. We initialize the centroids of the clusters to the vectors representing the senses from WordNet, either using their definition vectors $\mathbf{d}_{tj}$ in one series of experiments, or their example vectors $\mathbf{e}_{tj}$ in the other one. This second modification attempts to provide a reasonably accurate starting point for the clustering process.

3. After running the k-means algorithm, we reduce the number of clusters for each word type by merging the clusters which contain fewer than 10 tokens with the nearest larger cluster. This is done by calculating the cosine similarity between each token vector $\mathbf{u}_i$ and the centroids of the larger clusters and assigning the tokens to the closest large cluster. This re-clustering adapts the final number of clusters to the observed occurrences in the training data. Indeed, when there are few occurrences of a sense for a given ambiguous word type in the data, the SMT is likely not able to translate them properly due to the lack of training samples.

Finally, after clustering the training data, we use the centroids to assign each *new token from* *the test data* to a cluster, i.e. an abstract sense label, by selecting the closest centroid to it in terms of cosine distance in the embedding space.

7.1.4 Chinese Restaurant Process

The Chinese restaurant process (CRP) is an unsupervised method considered as a practical interpretation of a Dirichlet process [Ferguson, 1973] for non-parametric clustering. In the original analogy, each token is compared to a customer in a restaurant, each cluster being a table where customers are seated. When a new customer arrives, she can choose to sit at a table with other customers, with a probability proportional to the numbers of customers at that table, or sit at a new, empty table. In an application to multi-sense word embeddings, Li and Jurafsky [2015] proposed that the probability to "sit at a table" also depends on the contextual similarity between the token and the sense modeled by the table. We build upon this idea and bring several modifications that allow for an instantiation with sense-related knowledge from WordNet, as follows.

Here, for each word type W_t appearing in the data, we start by fixing the maximal number k_t of senses or clusters as the number of senses of W_t in WordNet. This avoids an unbounded number of clusters (as in the original CRP algorithm) and the risk of cluster sparsity, by setting a non-arbitrary limit based on linguistic knowledge. Moreover, we define the initial centroid of each cluster as the word vector corresponding either to the definition of the respective sense $\mathbf{d}_{tj}$, or alternatively to the example illustrating the sense $\mathbf{e}_{tj}$.

For each token w_i and its context vector $\mathbf{u}_i$, the algorithm decides whether the token is assigned to one of the sense clusters S_j to which previous tokens have been assigned, or whether it is assigned to an empty cluster, by selecting the option which has the highest probability, computed as follows:

$$P(w_i \to S_j) \propto \begin{cases} N_j(\lambda_1 s(\mathbf{u}_i, \mathbf{d}_{tj}) + \lambda_2 s(\mathbf{u}_i, \boldsymbol{\mu}_j)) & \text{if } N_j \neq 0 \text{ (non-empty sense)} \\ \gamma s(\mathbf{u}_i, \mathbf{d}_{tj}) & \text{if } N_j = 0 \text{ (empty sense)} \end{cases} \tag{7.3}$$

For a non-empty sense, the probability is proportional to the popularity of the sense (number of tokens it already contains, N_j) and to a weighted sum of two cosine similarities $s(\cdot,\cdot)$: one between the context vector $\mathbf{u}_i$ of the token and the definition of the sense $\mathbf{d}_{tj}$, and another one between $\mathbf{u}_i$ and the average context vector of the tokens already assigned to the sense ($\boldsymbol{\mu}_j$).These terms are weighted by the two hyper-parameters λ_1 and λ_2. For an empty sense, only the second term is used, weighted by the γ hyper-parameter.

Therefore, in our adaptive CRP approach, each token is assigned a sense based on:

1. the similarity between the token's context vector and the centroid vector of tokens in sense S_j noted $s(\mathbf{u}_i, \mu_j)$,

2. the similarity between the token's context vector and the pre-defined prior information of sense S_j noted $s(\mathbf{u}_i, \mathbf{d}_{tj})$,

3. the popularity of sense S_j, i.e. N_j.

We give the detailed explanation for each step of our adaptive CRP algorithm in Algorithm 1.

Algorithm 1 Adaptive Chinese Restaurant Process

```
repeat
    For each word type $W_t$ initialize prior probability to $\mathbf{d_{tj}}$ for each sense $S_j$ based on information from WordNet
    for Token $w_i$ do
        Calculate probabilities $P(w_i \rightarrow S_j)$ for each sense $S_j$ based on Eq. 7.3
        Select sense label $S_j$ to maximize $P(w_i \rightarrow S_j)$ and assign $w_i$ to it
        Update $\mu_j$ based on the assignment of $w_i$ to $S_j$
    end for
until All tokens of type $W_t$ have been analyzed
```

7.1.5 Random Walks

Finally, we also consider for comparison a WSD method based on random walks on the WordNet knowledge graph [Agirre and Soroa, 2009, Agirre et al., 2014] available within the UKB toolkit.[6] In the graph, senses correspond to nodes and the relationships or dependencies between pairs of senses correspond to the edges between those nodes.

From each input sentence, we extract its content words (nouns, verbs, adjectives, and adverbs) that have an entry in the WordNet weighted graph. The method calculates the probability of a random walk over the graph from a target word's sense ending on any other sense in the graph, and determines the sense with the highest probability for each analyzed word. In this case, the random walk algorithm is PageRank [Grin and Page, 1998], which computes a relative structural importance or 'rank' for each node in the graph. A link from node i to node j corresponds to a vote from node i to node j, which increases the rank of node j. In addition, the strength of this vote is proportional to the rank of node i.

Specifically, we first insert the context words into the graph G as nodes, and link them with directed edges to their respective concepts. Then we compute the personalized PageRank of the graph G by concentrating the initial probability mass uniformly over the newly introduced word nodes. As the words are linked to the concepts by directed edges, they act as source nodes injecting mass into the concepts they are associated with, which thus become relevant nodes, and spread their mass over the LKB graph. Therefore, the resulting the PageRank vector can be seen as a measure of the structural relevance of LKB concepts in the presence of the input context.

[6] http://ixa2.si.ehu.es/ukb/

7.2 Results and Analysis

We evaluate in this section our WSD systems on the dataset from the SemEval 2010 shared task [Manandhar et al., 2010], to assess their competitiveness, while acknowledging that our system uses external knowledge not available to SemEval participants. In this task, the target word dataset consists of 100 words: 50 nouns and 50 verbs.

	Training set	**Testing set**	**Senses (#)**
All	879807	8915	3.79
Nouns	716945	5285	4.46
Verbs	162862	3630	3.12

Table 7.1 – Training and testing sets provided for the SemEval 2010 task.

Table 7.1 shows the total number of target word instances in the training and testing sets, as well as the average number of senses in the gold standard. In this task, the training and testing data are treated separately, i.e. the testing data are only used for sense tagging, while the training data are only used for sense induction. Treating the testing data as new unseen instances ensures a realistic evaluation that allows to evaluate the clustering model.

7.2.1 Description of Evaluation Measures

The evaluation framework of SemEval-2010 considers two types of evaluation methods: (i) V-Measure [Rosenberg and Hirschberg, 2007] and (ii) Paired F-Score. In the evaluation, induced sense are mapped to gold standard ones using a mapping corpus. We explain below these two methods.

V-Measure

Let w be a target word with N instances in the testing dataset. Let $K = C_j | j = 1...n$ be a set of automatically generated clusters grouping theses instances, and $S = G_i | i = 1...m$ the set of gold standard classes containing the desirable groupings of w instances.

The V-Measure assesses the quality of a clustering solution by explicitly measuring its *homogeneity* and its *completeness*. Homogeneity refers to the degree that each cluster consists of data points primarily belonging to a single gold-standard class, while completeness refers to the degree that each gold-standard class consists of data points primarily assigned to a single cluster. Let h be homogeneity and c the completeness. V-Measure is the harmonic mean of h and c, i.e. $VM = \frac{2 \times h \times c}{h+c}$.

The homogeneity h is defined as 7.4:

$$h = \begin{cases} 1 & \text{if } H(S) = 0, \\ 1 - \frac{H(S|K)}{H(S)} & \text{otherwise} \end{cases} \tag{7.4}$$

where $H(S|K)$ is the conditional entropy of the class distribution, given the proposed clustering, and $H(S)$ is the class entropy.

When $H(S|K)$ is 0, the solution is perfectly homogeneous, because each cluster only contains data that belong to a single class. However in an imperfect situation, $H(S|K)$ depends on the size of the dataset and the distribution of class sizes. Hence, instead of taking the raw conditional entropy, V-Measure normalizes it by the maximum reduction in entropy the clustering information could provide, i.e. $H(S)$. When there is only a single class ($H(S) = 0$), any clustering would produce a perfectly homogeneous solution.

Symmetrically to homogeneity, the completeness c is defined in Eq. 7.5:

$$K = \begin{cases} 1 & \text{if } H(S) = 0, \\ 1 - \frac{H(K|S)}{H(K)} & \text{otherwise .} \end{cases} \tag{7.5}$$

where $H(K|S)$ is the conditional entropy of the cluster distribution given the class distribution and $H(K)$ is the clustering entropy. When $H(K|S)$ is 0, the solution is perfectly complete, because all data of a class belong to the same cluster.

Paired F-Score

In this measure, the clustering problem is transformed into a classification problem. For each cluster C_i we generate $\begin{pmatrix} |C_i| \\ 2 \end{pmatrix}$ instance pairs, where $|C_i|$ is the total number of instances that belong to cluster C_i. Similarly, for each gold-standard class G_i we generate $\begin{pmatrix} |G_i| \\ 2 \end{pmatrix}$ instance pairs, where $|G_i|$ is the total number of instances that belong to gold-standard class G_i.

Let $F(K)$ be the set of instance pairs that exist in the automatically induced clusters and $F(S)$ be the set of instance pairs that exist in the gold standard. Precision can be defined as the number of common instance pairs between the two sets to the total number of pairs in the clustering solution as shown in Eq. 7.6, while recall can be defined as the number of common instance pairs between the two sets to the total number of pairs in the gold standard in Eq 7.6. Finally, precision and recall are combined to produce the harmonic mean F_1.

$$\text{Precision} = \frac{|F(K)| \cap F(S)}{|F(K)|} \qquad \text{Recall} = \frac{|F(K)| \cap F(S)}{|F(S)|} \tag{7.6}$$

7.2.2 Evaluation Scores and Analysis

Table 7.2 shows our WSD results in terms of V-score and F_1-score, comparing our methods (bottom six lines) with other significant systems that participated in the SemEval 2010 shared task [Manandhar et al., 2010]. We add three baselines provided by the task organizers for comparison:

1. Most Frequent Sense (MFS), which groups all occurrences of a word into one cluster;

2. 1ClusterPerInstance, which produces one cluster for each occurrence of a word;

3. Random, which randomly assigns an occurrence to 1 out of 4 clusters (4 is the average number of senses from the ground-truth).

	System	V-score			F_1-score		
		All	Nouns	Verbs	All	Nouns	Verbs
Base.	MFS	0	0	0	64.85	57.00	72.70
	Random	4.40	4.60	4.20	32.35	30.60	34.10
	1ClusterPerIns	31.70	35.80	25.60	0.12	0.11	0.12
	Hermit [Jurgens and Stevens, 2010]	16.20	16.70	15.60	25.55	26.70	24.40
Top systems	UoY [Korkontzelos and Manandhar, 2010]	15.70	20.60	8.50	49.80	38.20	66.60
	KSU KDD [Elshamy et al., 2010]	15.70	18.00	12.40	36.90	24.60	54.70
	Duluth-WSI [Pedersen, 2010]	9.00	11.40	5.70	41.10	37.10	46.70
	Duluth-WSI-SVD-Gap [Pedersen, 2010]	0.00	0.00	0.10	63.30	57.00	72.40
	KCDC-PT [Kern et al., 2010]	1.90	1.00	3.10	61.80	56.40	69.70
	KCDC-GD [Kern et al., 2010]	6.90	5.90	8.50	59.20	51.60	70.00
	Duluth-Mix-Gap [Pedersen, 2010]	3.00	2.90	3.00	59.10	54.50	65.80
Ours	***k*-means + definition**	13.65	14.70	12.60	56.70	53.70	59.60
	k-means + example	11.35	11.00	11.70	53.25	47.70	58.80
	CRP + definition	1.45	1.50	1.45	64.80	56.80	72.80
	CRP + example	1.20	1.30	1.10	64.75	56.80	72.70
	Graph + definition	11.30	11.90	10.70	55.10	52.80	57.40
	Graph + example	9.05	8.70	9.40	50.15	45.20	55.10

Table 7.2 – WSD results from the SemEval 2010 shared task in terms of V-score and F_1-score. Our adaptive k-means using definitions (last but one line) outperforms all the other systems on the average of V and F_1, when considering both nouns and verbs, or nouns only.

The V-score is biased towards system generating a higher number of clusters than the number of gold standard senses. F_1-score measures the classification performance, i.e. how well a method assigns two occurrences of a word belonging to the same gold standard class. Hence, this metric favors systems that generate fewer clusters: if all instances were grouped into 1 cluster, the F_1-score would be high. For instance, the MFS baseline system, which assigns all occurrences into one cluster, presents a high F_1 score but obtains a 0 score on the V-measure. Therefore, only considering the results of a single evaluation method, V-measure or F_1, would lead to biased conclusions.

	System	Average			
		All	Nouns	Verbs	#clusters
Base.	MFS	32.42	29.50	25.40	1.00
	Random	18.45	17.60	19.30	4.00
	1ClusterPerIns	15.40	17.90	12.90	89.15
Top systems	Hermit [Jurgens and Stevens, 2010]	20.85	21.70	20.00	10.78
	UoY [Korkontzelos and Manandhar, 2010]	32.75	29.40	**37.50**	11.54
	KSU KDD [Elshamy et al., 2010]	26.30	21.30	33.50	17.50
	Duluth-WSI [Pedersen, 2010]	25.05	24.20	26.20	4.15
	Duluth-WSI-SVD-Gap [Pedersen, 2010]	31.65	28.50	36.20	1.02
	KCDC-PT [Kern et al., 2010]	31.85	28.70	36.40	1.50
	KCDC-GD [Kern et al., 2010]	33.05	28.70	39.20	2.78
	Duluth-Mix-Gap [Pedersen, 2010]	31.05	29.70	34.40	1.61
Ours	***k*-means + definition**	**35.20**	**34.20**	36.10	4.45
	k-means + example	32.28	29.30	35.25	3.58
	CRP + definition	33.13	29.15	37.10	1.80
	CRP + example	32.98	29.05	36.90	1.66
	Graph + definition	33.20	32.35	34.05	2.63
	Graph + example	29.60	29.96	32.25	2.08

Table 7.3 – WSD results from the SemEval 2010 shared task in terms of the average value of V-score and F_1 score . Our adaptive k-means using definitions (last but one line) outperforms all the other systems on the average of V and F_1, when considering both nouns and verbs, or nouns only.

As these two metrics are biased towards either small or large numbers of clusters, we calculate the mean value of the two scores generated from each system (V-score and F_1 score) and present them in Table 7.3. The results show that the adaptive k-means initialized with definitions has the highest average score (35.20) and ranks among the top systems for most of the metrics individually. Moreover, the adaptive k-means method finds on average 4.5 senses per word type, which is very close to the ground-truth value provided by SemEval (4.46). Overall, we observed that k-means infers fewer senses per word type than WordNet. These results show that our method is effective and provides competitive performance against prior art, partly thanks to additional knowledge not available to the shared task system.

This chapter presented three adaptive context-dependent clustering algorithms for WSD: k-means, Chinese Restaurant Process and Random Walk. All these algorithms utilized semantic information from WordNet to identify the dominant clusters, which correspond to senses in the source side of a parallel corpus. The proposed adaptive k-means method provides competitive WSD performance on data from the SemEval 2010 shared task. We will integrate the resulting sense information with SMT in the next Chapter 8, and later use it to initialize sense embeddings for NMT in Chapter 9.

8 Sense-Aware Statistical Machine Translation

In the previous chapter, we designed a WSD system that determines the senses of ambiguous words thanks to the use of an external resource (WordNet). In this chapter, we introduce a sense-aware SMT system that uses the proposed adaptive WSD. The labels of the clusters resulting from WSD are used as abstract source-side sense labels within a factored phrase-based SMT system, presented in Section 8.3. Our results, presented in Section 8.4, show that the adaptive WSD approach helps SMT to increase its BLEU scores and to improve the translation of polysemous nouns and verbs, when translating from English into Chinese, German, French, Spanish or Dutch, in comparison to an SMT baseline that is not aware of word senses.

Current statistical MT system performs WSD implicitly, for instance through the n-gram frequency information stored in the translation and language models. However, the context taken into account by an MT system when performing implicit WSD is limited. In the case of phrase-based SMT, the size of the context is related to the order of the language model (often between 3 and 5) and the length of n-grams in the phrase table (seldom above 5).

The example below shows an English sentence translated into German by a baseline statistical MT, and by the sense-aware SMT system proposed in this chapter. The word *shot* is respectively translated as *Bild* (drawing) by the baseline system, and as *Aufnahme* by our sense-aware SMT, which selects a correct sense and a word that is identical to the reference translation, unlike the baseline system which is mistaken.

Source: And I do really like this *shot*, because it shows all the detritus that's sort of embedded in the sole of the sneakers.

Baseline SMT: Und ich mag dieses *Bild* ...

Sense-aware SMT: Und ich mag diese *Aufnahme* wirklich, ...

Reference translation: Ich mag diese *Aufnahme* wirklich, ...

8.1 Datasets, Preparation and Settings

We evaluate our sense-aware SMT on the UN Corpus[1] [Rafalovitch and Dale, 2009] and on the Europarl Corpus[2] [Koehn, 2005]. We select 0.5 million parallel sentences for each language pair from Europarl, as shown in Table 8.1. We also use the smaller WIT3 Corpus[3] [Cettolo et al., 2012], with transcripts of TED talks, to evaluate the impact of costly model choices, namely the type of the sense-related knowledge (definition vs. examples), the length of the context window, and the k-means method (adaptive vs. baseline).

		Training		Development		Testing		Labels		Words
		Lines	Tokens	Lines	Tokens	Lines	Tokens	Nouns	Verbs	
EN/ZH	WIT3	150K	3M	10K	0.3M	50K	1M	2659	1440	1320
	UN	500K	13M	5K	0.14M	50K	1.5M	3550	1748	1942
EN/DE	WIT3	140K	2.8M	5K	0.16M	50K	1M	2186	1049	1294
	Europarl	500K	14M	5K	0.14M	50K	1.4M	3885	1576	1976
EN/FR	Europarl	~	~	~	~	~	~	3910	1627	2006
EN/ES	Europarl	~	~	~	~	~	-	3862	1627	1987
EN/NL	Europarl	~	~	~	~	~	~	3915	1647	2210

Table 8.1 – Statistics of the corpora used for machine translation: '~' indicates a similar size, but not identical texts, because the English source texts for the different language pairs from Europarl are different. Hence, the number of words found in WordNet differ as well.

Before assigning sense labels, we first tokenize all the texts and identify the parts of speech (POS) using the Stanford POS tagger[4]. Then, we filter out the stopwords and the nouns which are proper names according to the Stanford Name Entity Recognizer[4]. Furthermore, we convert the plural forms of nouns to their singular form and the verb forms to infinitive using the stemmer and lemmatizer from NLTK[5], which is essential because WordNet has entries only for singular nouns and infinitive forms of verbs. The pre-processed text is used for assigning sense labels to each occurrence of a noun or verb which has more than one sense in WordNet.

Our adaptive WSD system assigns a sense number for each ambiguous word token in the source-side of a parallel corpus. To pass this information to an SMT system, we use a factored phrase-based translation model [Koehn and Hoang, 2007]. The factored model offers a principled way to supplement words with additional information – such as, traditionally, part-of-speech tags – without requiring any intervention in the translation tables. The features are combined in a log-linear way with those of a standard phrase-based decoder, and the goal remains to find the most probable target sentence for a given source sentence as follows:

$$f = \arg\max_{f} \exp \sum_{i=1}^{n} \lambda_i h_i(e, f) \qquad (8.1)$$

[1] http://www.uncorpora.org/

[2] http://www.statmt.org/europarl/

[3] http://wit3.fbk.eu/

[4] http://nlp.stanford.edu/software/

[5] http://www.nltk.org/

where n is the number of features, $h_i(e, f)$ is the feature function for factor i, and λ_i is the weight for that factor, which are optimized during tuning. To each source noun or verb token, we add a sense label obtained from our adaptive WSD system. To all the other words, we add a NULL label. In practice, these labels are simply appended to the tokens in the data with a vertical bar, e.g. 'rock|1' or 'great|NULL'. The translation system will thus take the source-side sense labels into consideration during the training and the decoding processes. Hence, the feature functions depend on the source sentence e which contains words and sense labels, and the target vector f which contains only words.

We select the optimal model configuration based on the MT performance, measured with the traditional BLEU score [Papineni et al., 2002], on the WIT3 corpus for EN/ZH and EN/DE. Unless otherwise stated, we use the following settings in the k-means algorithm:

- we use the definition of each sense for initializing the centroids in the adaptive k-means methods (and compare this later with using the examples);
- we set k_t equal to m_t, i.e. the number of senses of an ambiguous word type W_t;
- the window size for the context surrounding each occurrence is set to $c = 8$.

To measure the impact of WSD on MT, besides BLEU, we also measure the actual impact on the nouns and verbs that appear in WordNet with several senses, by comparing how many of them are translated as in the reference translation, by our system vs. by the baseline system.

For a certain set of tokens in the source data, we note as N_{improved} the number of tokens which are translated by our system as in the reference translation, but whose baseline translation differs from it. Conversely, we note as N_{degraded} the number of tokens which are translated by the baseline system as in the reference, but differently by our system. We will use the normalized coefficient $\rho = (N_{\text{improved}} - N_{\text{degraded}})/T$, where T is the total number of tokens, as a metric focusing explicitly on the words submitted to WSD.[6]

8.2 Optimal Values of the Parameters

Using the data, settings, and metrics in Section 8.1, we investigate first the impact of two model choices on the performance: centroid initialization for k-means (definitions, examples, or random), and the length of the context window for each word. Then, we evaluate our adaptive WSD+MT performance.

[6] The values of N_{improved} and N_{degraded} are obtained using automatic word alignment. They do not capture, of course, the absolute correctness of a candidate translation, but only its identity or not with one reference translation.

8.2.1 Initialization of Adaptive k-means

We examine first the impact of the initialization of the sense clusters, on the WIT3 Corpus. In Table 8.2, we present the BLEU scores of our WSD+MT system in two conditions: when the k-means clusters are initialized with vectors from the definitions vs. from the examples provided in the WordNet synsets of ambiguous words. Moreover, we provide BLEU scores of baseline systems and oracle ones (i.e. using correct senses as factors), as well as the ρ score indicating the relative improvement of ambiguous words in our system with respect to the baseline.

The use of definitions outperforms the use of examples, probably because there are more words with definitions than with examples in WordNet (twice as many, as shown in Table 8.1 in Section 8.1), but also because definitions may provide more helpful words to build the initial vectors, as they are more explicit than the examples. All the values of the ρ coefficient show clear improvements over the baseline, up to 4% for DE/EN. As for the oracle scores, they outperform the baseline by a factor of 2–3 compared to our system.

Pair	Resource	BLEU			ρ (%)
		Baseline	Factored	Oracle	
EN/ZH	Definitions	15.23	**15.54**	16.24	+2.25
	Examples		15.41	15.85	+1.60
EN/DE	Definitions	19.72	**20.23**	20.99	+3.96
	Examples		19.98	20.45	+2.15

Table 8.2 – Performance of our WSD+MT factored system for two language pairs from WIT3, with two initialization conditions for the k-means clusters, i.e. definitions or examples for each sense.

In addition, we compare the two initialization options above with random initializations of k-means clusters, in Table 8.3. To offer a fair comparison, we set the number of clusters, in the case of random initializations, respectively to the number of synsets with definitions or examples, for each word type. We obtain BLEU scores of 15.34 and 15.27 respectively, on EN/ZH – hence lower than 15.54 and 15.41 in Table 8.2, showing that our adaptive and informed initializations of clusters are beneficial to MT.

Resource	k-means initialization	
	Specific	Random
Definitions	**15.54**	15.34
Examples	**15.41**	15.27

Table 8.3 – Performance of our WSD+MT factored system for EN-ZH from WIT3, comparing the two initialization conditions for the k-means clusters, i.e. definitions or examples for each sense, with random initializations.

8.2.2 Length of the Context Window

We investigate the effect of the size of the context window surrounding each ambiguous token, i.e. the number of words surrounding it that are considered for building its vector representation. Figure 8.1 displays the BLEU score of our WSD+MT factored system when varying this size, on EN/ZH translation in the WIT3 Corpus, along with the (constant) score of the baseline. The performance of our system improves with the size of the window, reaching a peak around 8–10 words. This result highlights the importance of a longer context compared to the typical settings of SMT systems, which generally do not go beyond 6 words.

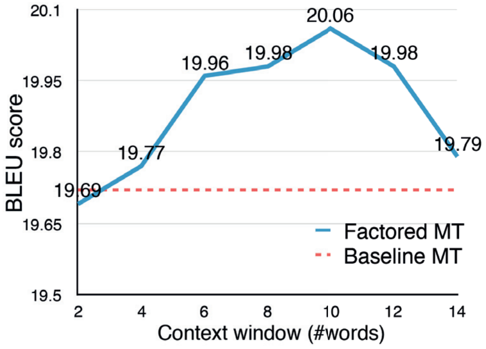


Figure 8.1 – BLEU scores of our WSD+MT factored system on EN/ZH WIT3 data, along with the baseline score (constant), when the size of the context window around each ambiguous token (for building its context vector) varies from 2 to 14 words.

It also suggests that MT systems which exploit effectively longer context, as we show here with a sense-aware factored MT system for ambiguous nouns and verbs, can significantly improve their lexical choice and their overall translation quality.

8.3 Integration with Statistical MT

Our adaptive WSD system assigns a sense number to each ambiguous word token (noun or verb) in the source-side of a parallel corpus. To pass this information to an SMT system, we use a factored phrase-based translation model [Koehn and Hoang, 2007]. The factored model offers a principled way to supplement words with additional information – such as, traditionally, part-of-speech tags – without requiring any intervention in the translation tables. The features are combined in a log-linear way with those of a standard phrase-based decoder, and the goal remains to find the most probable target sentence for a given source sentence.

8.4 Results and Analysis

Table 8.4 shows the BLEU scores of our factored system with sense label integration. We compare the translation performance with labels generated from the three different clustering methods studied in the previous chapter. Our systems perform consistently better than the baseline with all clustering methods on all language pairs. The system with labels obtained from k-means consistently outperforms the other two systems. So we select the k-means method for our further experiments and analyses.

Language pair	BLEU			
	Baseline	Graph	CRP	k-means
EN/ZH	23.25	23.47 *(+.22)*	23.55 *(+.29)*	23.69 *(+.44)*
EN/DE	20.78	21.17 *(+.39)*	21.19 *(+.41)*	21.32 *(+.54)*
EN/FR	31.96	32.01 *(+.05)*	32.08 *(+.12)*	32.20 *(+.24)*
EN/ES	39.95	40.15 *(+.20)*	40.14 *(+.19)*	40.37 *(+.42)*
EN/NL	23.56	23.74 *(+.18)*	23.79 *(+.23)*	23.84 *(+.26)*

Table 8.4 – BLEU scores of the WSD+MT factored system with three clustering methods on five language pairs.

Language pair	Corpus	BLEU			ρ (%)
		Baseline	Factored	Oracle	
EN/ZH	UN	23.25	**23.69**	24.44	+2.26
EN/DE	Europarl	20.78	**21.32**	21.95	+1.57
EN/FR	Europarl	31.96	**32.20**	32.98	+1.21
EN/ES	Europarl	39.95	**40.37**	41.06	+1.04
EN/NL	Europarl	23.56	**23.84**	24.79	+1.38

Table 8.5 – BLEU scores of our WSD+MT factored system, with both noun and verb senses, along with baseline MT and oracle WSD+MT, on five language pairs.

In order to evaluate the translation quality of the labeled words, we analyze separately the nouns vs. the verbs. Table 8.5 displays the performance of our factored MT systems (with k-means clustering for WSD) trained with noun and verb senses on five language pairs, using the datasets specified in Table 8.1. Our system performs consistently better than the MT baseline on all pairs, with the largest improvements achieved on EN/ZH and EN/DE. To better understand the improvements over the baseline MT, we also provide the BLEU score of an oracle system which has access to the reference translation of the ambiguous words, which we obtain using the alignment provided by GIZA++. According to the results shown in Table 8.5, our factored MT system bridges around 40% of the gap between the baseline MT system and the oracle system on EN/DE and 30% on EN/ZH, and shows similar improvements for the other language pairs as well.

Lastly, Table 8.6 shows the confusion matrix for our factored MT integrated with sense labels from k-means and the baseline MT systems when comparing the reference translations of nouns and verbs separately, using GIZA++ alignments. The confusion matrix displays the number of labeled tokens which are translated as in the reference or not ('Correct' vs. 'Incorrect'). As we can observe, the number of tokens that our factored MT system translates correctly while the baseline MT does not, is two times larger than the number of tokens that the baseline MT system translates correctly while our factored MT does not. These increments

		Factored (Nouns)				Factored (Verbs)			
		nouns		nouns + verbs		verbs		nouns + verbs	
		Cor.	Inc.	Cor.	Inc.	Cor.	Inc.	Cor.	Inc.
EN/ZH	Correct	138,876	4,402	138,264	5,075	37,132	1,166	36,647	1,527
Baseline	Incorrect	8,454	75,690	9,472	74,541	3,939	41,728	4,149	41,077
EN/DE	Correct	91,966	1,473	91,376	2,035	18,370	664	18,214	812
Baseline	Incorrect	4,268	71,037	4,525	69,931	1,892	47,105	2,029	46,795
EN/FR	Correct	128,551	3,010	128,142	3,371	29,310	1,109	29,099	1,305
Baseline	Incorrect	4,964	79,886	5,142	79,632	2,983	58,890	3,266	58,483
EN/ES	Correct	151,123	1,933	151,092	1,941	57,610	1464	58,596	618
Baseline	Incorrect	4,156	53,748	3,975	53,879	4,340	74,922	2,219	77,355
EN/NL	Correct	102,323	1,445	102,356	1,503	16,336	776	15,973	987
Baseline	Incorrect	3,496	73,613	3,981	73,444	1,640	51,514	1,707	50,911

Table 8.6 – Detailed confusion matrix of our factored MT system and the baseline MT system with respect to the reference on all language pairs from Europarl corpus and UN corpus.

are consistent for all the five language pairs under study.

In this chapter, we presented a sense-aware statistical MT system which uses a larger context than standard ones, thanks to an adaptive WSD method explained in Chapter 7. We experimented with five language pairs and showed that our sense-aware MT method consistently improves over the baseline. We will integrate sense information for ambiguous words to neural MT in the next Chapter 9.

9 Neural Machine Translation with Sense Knowledge Integration

We showed in the previous chapter that sense-aware SMT outperforms an SMT baseline that does not have access to sense information. In this chapter, we study several approaches to NMT that integrate word sense information. In the baseline attention-based NMT approach, the context can extend to the entire sentence, but multiple word senses are not modeled explicitly. This is because all the occurrences of a word type share the same word representation in the translation process, and NMT systems do not explicitly model the contexts in which one word sense should be used instead of another one. For instance, in the sentence below from Europarl, the translation of 'deal' should convey the sense 'to handle' (in French '*traiter*') and not 'to cope' (in French '*remédier*'):

Source: How can we guarantee the system of prior notification for high-risk products at ports that have the necessary facilities to *deal* with them?

Reference translation: Comment pouvons-nous garantir le système de notification préalable pour les produits présentant un risque élevé dans les ports qui disposent des installations nécessaires pour *traiter* ces produits ?

Baseline NMT: Comment pouvons-nous garantir le système de notification préalable pour les produits à haut risque dans les ports les ports qui disposent des moyens nécessaires pour y *remédier* ?

Sense-aware NMT: Comment garantir le système de notification préalable des produits à haut risque dans les ports qui disposent des installations nécessaires pour les *traiter* ?

In this chapter, we demonstrate that the explicit modelling of word senses can be helpful to NMT by using combined vector representations of word types and word senses, which are inferred from contexts that are larger than those of state-of-the-art NMT systems. Concretely, we make the following contributions:

- We propose three sense selection mechanisms for integrating into NMT sense knowledge (modeled as in Chapter 7) respectively based on top, average, and weighted average – i.e. attention – of word senses (see Section 9.1).

- We achieve consistent improvements against baseline NMT on five language pairs: from English into Chinese, German, French, Spanish and Dutch (see Section 9.2).

9.1 Sense Selection Mechanisms

In this section, we present our new models that integrate sense information into neural machine translation. The proposed models concatenate the word embedding $\mathbf{w}_i$ of each token w_i with a vector representation of its hypothesized sense μ_i, which is either obtained from one of the clustering methods presented in Chapter 7, or learned during encoding, as will be explained in the rest of this section.

For better illustration, we start with an example source sentence, which contains an ambiguous word 'rock':

I love **rock** music very much.

In our model, we will represent this sentence with label specification on the ambiguous word as follows:

I love **rock**|*sense* music very much.

In the baseline NMT system, a word vocabulary (noted below as WV) for each language side is first created, which contains the top N unique words that appear in the training documents (ranked by their frequency). We set N to $50,000$ in our study. The input to the encoder is the vector representation $\mathbf{w}_i$ of each word token w_i from the word vocabulary. The embeddings of the words are learned in the training process. For instance, for the example above, we note $\mathbf{w}_{rock}$ the word embedding that is fed into the encoder when the source sentence is analyzed.

However, in our study, we introduce a new vector named Sense Embedding (noted μ_i) to represent the sense information of each word token w_i in the training data. In the training process, we provide the concatenation of each token's word embedding $\mathbf{w}_i$ and its corresponding sense embedding μ_i as input to the encoder. We then let the system optimize the two embeddings in parallel. Taking 'rock' as example, the actual embedding input to the encoder in our study becomes $\mathbf{w} = [\mathbf{w}_{rock}; \mu_{rock}]$.

To generate the sense embedding of each word, we need an additional vocabulary which contains not only all the words from the WV, but also all the sense labels, named Sense-Label Vocabulary (noted SLV). For the sample source sentence given above (also shown in Figure 9.1), the WV is ['I', 'love', 'rock', 'music', 'very', 'much'], and the SLV is ['I', 'love', 'rock', 'music', 'very', 'much', 'r_1'].

For those words that do not have sense labels, the sense embedding is simply the embedding of the word with respect to the SLV. For the words that have sense labels, we propose several different models to construct and compute their sense embeddings μ_i for each ambiguous token w_i, described as follows.

Top sense (*TOP*)

In this first model, for each ambiguous word, we directly use the word's label embedding with respect to the SLV as sense embedding. Here the sense label of a particular word token in the document is determined by our adaptive WSD system presented in Chapter 7.

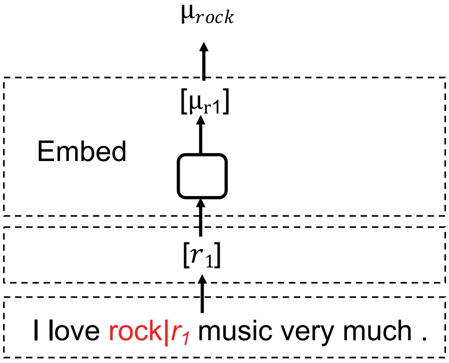


Figure 9.1 – Generation of sense embedding μ_{rock} by the *TOP* method, using as an example the ambiguous token 'rock'.

Figure 9.1 illustrates the sense embedding of 'rock' in the given example sentence. Assuming that our WSD system estimates the sense label of word 'rock' as 'r_1', we use the embedding μ_{r_1} of 'r_1' with respect to the SLV as the sense embedding $\mu_{rock} = \mu_{r_1}$ in this sentence.

Weighted average of senses (*AVG)*

In this model, we consider all possible sense labels of each ambiguous word instead of directly trusting the decision of the WSD system.

In learning our WSD system using k-means, we obtain the cluster centroids of the word senses and convert the distances d_l between the input token vector and the centroid of each sense S_l into a normalized weight distribution, either by a linear or a logistic normalization:

$$\omega_j = \frac{1 - d_j}{\sum_{1 \le l \le k} d_l} \tag{9.1}$$

or

$$\omega_j = \frac{e^{-d_j^2}}{\sum_{1 \le l \le k} e^{-d_l^2}} \tag{9.2}$$

The sense embedding μ_i for each ambiguous token w_i is then computed as the weighted average of all sense label embeddings:

$$\boldsymbol{\mu}_i = \sum_{1 \le j \le k} \omega_j \boldsymbol{\mu}_{ij} \tag{9.3}$$

Figure 9.2 represents the generation of the sense embedding μ_{rock} based on this model. In this figure, we assume that 'rock' has only two possible senses: r_1 and r_2. We first obtain the sense label embeddings of the two sense labels (r_1 and r_2) with respect to SLV and then construct the final sense embedding μ_{rock} of the word 'rock' as the weighted average of the two label vectors (μ_{r_1} and μ_{r_2}). The weights of the two labels are obtained as the distance between 'rock' and the centroid of the correlated sense labels (r_1 and r_2). Since we generate the weight distribution for each ambiguous token from the WSD system, the distributions are fixed and will not be optimized during encoding process.

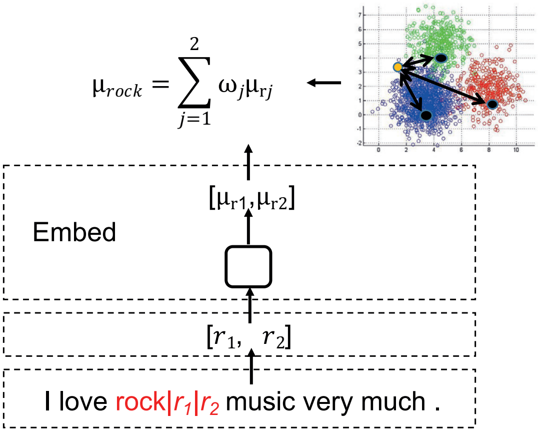
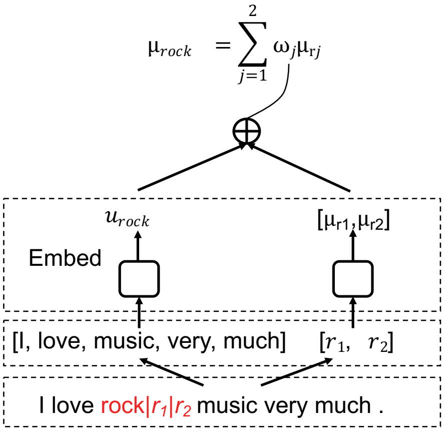

$\mu_{rock} = \sum_{j=1}^{2} \omega_j \mu_{rj}$
Embed
$[\mu_{r1}, \mu_{r2}]$
$[r_1, \; r_2]$
I love rock|r_1|r_2 music very much .

Figure 9.2 – Generation of sense embedding μ_{rock} by the *AVG* method, using as an example the ambiguous token 'rock'.

Attention-based sense weights (*ATT*) Following the *AVG* model, we propose a third model that computes the relatedness probability of each sense label dynamically instead of using fixed weights. The weights are generated based on the current word and sense label embeddings in every iteration of the training process, at the encoding stage.

Figure 9.3 shows the entire procedure of generating the sense embedding μ_{rock} for the 'rock' token in this model. We first collect the context words of 'rock', i.e. all the other words in the sentence. We define the context vector $\mathbf{u}_{rock}$ of 'rock' as the average of all the embeddings of its context words. Meanwhile, we consider the label embeddings μ_{r_1} and μ_{r_2} of the sense labels of 'rock' (here r_1 and r_2) with respect to SLV and compute the similarity between each label embedding (μ_{r_1} and μ_{r_1}) and the context vector $\mathbf{u}_{rock}$ for 'rock', using an additional attention



$\mu_{rock} = \sum_{j=1}^{2} \omega_j \mu_{rj}$
u_{rock}
$[\mu_{r1}, \mu_{r2}]$
Embed
[I, love, music, very, much]
$[r_1, r_2]$
I love rock|r_1|r_2 music very much .

Figure 9.3 – Generation of sense embedding μ_{rock} by the *ATT* method, using as an example the ambiguous token ‘rock’.

layer in the network. We compare the following two functions used in this attention layer:

$$f(\mathbf{u}_i, \boldsymbol{\mu}_{ij}) = v^T tanh(W\mathbf{u}_i + U\boldsymbol{\mu}_{ij}) \tag{9.4}$$

or

$$\mathbf{u}_i^T W \boldsymbol{\mu}_{ij} \tag{9.5}$$

The weights ω_j are now obtained through the following softmax normalization:

$$\omega_j = \frac{e^{f(\mathbf{u}_i, \boldsymbol{\mu}_{ij})}}{\sum_{1 \le l \le k} e^{f(\mathbf{u}_i, \boldsymbol{\mu}_{il})}} \tag{9.6}$$

Finally, the average sense embedding μ_{rock} is obtained as in Equation 9.3, and is concatenated to the word vector $\mathbf{w}_i$. Since the correlated weight distribution for each ambiguous token is calculated based on the current embeddings of words and sense labels, the parameters will be optimized during encoding process.

Model *ATT* with initialization of embeddings (*ATT*$_{ini}$)

The fourth model is a variation of the *ATT* model. The only difference is the way word embeddings are initialized with respect to WV and sense label embeddings with respect to SLV.

In this model we initialize the word embedding using the word2vec vectors of corresponding word types, and the sense label embeddings using the centroid vectors obtained from *k*-

means.

9.2 Experimental Settings, Results and their Analysis

9.2.1 Number of Senses to be Considered

In order to address the sparsity of senses that we considered, we present and compare the sense number distribution learned from either our WSD system or provided by WordNet on EN/FR translation in the Europarl Corpus. As shown in Figure 9.4, the distribution learned from WordNet is less convergent than the results learned from our adaptive k-means. In the sense-aware NMT, we integrate the sense vector by averaging the weights of all its possible senses. We set to 5 the maximum number of senses to be considered, since there are fewer than 100 words with more than 5 senses. We consider the most related 5 senses for the tokens which originally have more than 5 senses.

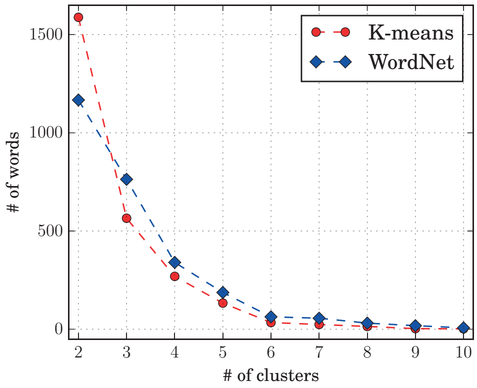


Figure 9.4 – Sparsity of the distribution of senses, where the X-axis is the distribution of the number of senses per token, while the Y-axis is the number of tokens which have the related number of senses.

9.2.2 Selection of WSD+NMT Model

To compare several options of the WSD+NMT systems, we trained and tested them on a subset of EN/FR Europarl.[1] The results are shown in Table 9.1. For the *AVG* model, the logistic normalization in Eq. 9.2 works better than the linear one in Eq. 9.1. For the *ATT* model, we compared two different labeling approaches for tokens which do not have multiple senses: either use the same NULL label for all tokens, or use the word itself as a label for its sense; the second option appeared to be the best. Finally, for the ATT_{ini} model, we compared the two options for the attention function in Eq. 9.4 and Eq. 9.5, and found that the Eq. 9.4 is the best. In what follows, we use these settings for the *AVG* and *ATT* systems.

[1] The smaller dataset allowed for shorter training times.

System and settings	BLEU
Baseline	29.55
TOP	29.63 *(+0.08)*
AVG with linear algorithm in Eq. 9.1	29.67 *(+0.12)*
AVG with logistic algorithm in Eq. 9.2	**30.15** *(+0.60)*
ATT with NULL label	29.80 *(+0.33)*
ATT with word used as label	**30.23** *(+0.68)*
ATT_{ini} in Eq. 9.5	29.94 *(+0.39)*
ATT_{ini} in Eq. 9.4	**30.61** *(+1.06)*

Table 9.1 – Performance of various WSD+NMT configurations on a EN/FR subset of Europarl, with variations wrt. baseline. We select the settings with the best performance (bold) for our final experiments in Section 9.2.3.

9.2.3 Neural Machine Translation Results

BLEU scores. The results of the MT systems trained on the full Europarl datasets (or UN, for EN/ZH) are presented in Table 9.2 for five language pairs with English as source. The best hyper-parameters are those found above, for each of the WSD+NMT combination strategies. The best scores are reached with ATT_{ini} for WSD+NMT, i.e. the attention-based model of senses initialized with the centroid vectors of k-means clustering and word embeddings from word2vec.

	EN/FR	EN/DE	EN/ZH	EN/ES	EN/NL
Baseline	34.60	25.80	27.07	44.09	24.79
k-means + *TOP*	34.52 *(−.08)*	25.84 *(+.04)*	26.93 *(−.14)*	44.14 *(+.05)*	24.71 *(−.08)*
k-means + *AVG*	35.17 *(+.57)*	26.47 *(+.67)*	27.44 *(+.37)*	45.05 *(+.97)*	25.04 *(+.25)*
None + *ATT*	35.32 *(+.72)*	26.50 *(+.70)*	27.56 *(+.49)*	44.93 *(+.84)*	25.36 *(+.57)*
k-means + ATT_{ini}	**35.78 *(+1.18)***	**26.74 *(+.94)***	**27.84 *(+.77)***	**45.18 *(+1.09)***	**25.65 *(+.86)***

Table 9.2 – BLEU scores of our sense-aware NMT systems over five language pairs: ATT_{ini} is the best one.

Comparisons with baselines. Table 9.2 shows that our WSD+NMT systems perform consistently better than the baselines, with the largest improvements achieved by NMT on EN/FR and EN/ES. The neural systems also outperform our previous results with phrase-based SMT presented in Chapter 8.

Lexical choice. Using word alignment, we assess the improvement brought by our systems with respect to a baseline in terms of the number of words – here, WSD-labeled nouns and verbs – that are translated exactly as in the reference translation (modulo alignment errors). These numbers can be arranged in a matrix (Table 9.3) displaying four numbers: the words translated correctly (i.e., as in the reference) by both systems, those translated correctly by one system but incorrectly by the other one, and vice-versa, and those translated incorrectly by both.

Table 9.3 shows this matrix for our sense-aware NMT with the ATT_{ini} model *versus* the NMT baseline over the Europarl test data. The net improvement, i.e. the fraction of words improved by our system minus those degraded, appears to be +2.5% for EN/FR and +3.6% for EN/ES. If we compare these results with those of the sense-aware SMT system shown in Table 8.6 (Chapter 8), the ATT_{ini} NMT model brings higher benefits over the NMT baseline than the WSD+SMT factored model, although the NMT baseline is stronger than the SMT one (Table 8.4).

		Baseline			
		EN/FR		EN/ES	
		Correct	Incorrect	Correct	Incorrect
ATT_{ini}	Correct	134,552	17,145	146,806	16,523
	Incorrect	10,551	101,228	8,183	58,387

Table 9.3 – Confusion matrix for our (a) WSD+NMT (ATT_{ini}) system against the baseline, over the Europarl test data, for different language pairs.

Human assessment. To compare our system against the baseline – apart from automatically counting the number of translations identical or not to the reference, as above – we also consider the human assessment of the translation of words with multiple senses (nouns or verbs). The goal is to capture more precisely correct translations that are, however, different from the reference.

Given the cost of the procedure, one evaluator with good knowledge of EN and FR rated the translations of four word types that appear frequently in the test set – ‘deal’ (101 tokens), ‘face’ (84), ‘mark’ (20) , and ‘subject’ (58) – and have multiple possible senses and translations into French (see also the example of ‘deal’ in the beginning of this chapter). For each occurrence, the evaluator saw the source sentence, the reference translation, and the outputs of the NMT baseline and the ATT_{ini} in random order, so that the system could not be identified. The two translations of the considered word were rated as good, acceptable, or wrong. This scale is thus not uniform, and we are not going to use it numerically below. Only cases in which the two translations differed were considered, to minimize the annotation effort with no impact on the comparison between systems.

Firstly, Fig. 9.5 shows that ATT_{ini} has a higher proportion of good translations, and a lower proportion of wrong ones, for all four words. The largest difference is for ‘subject’, where ATT_{ini} has 75% good translations and the baseline only 46%; moreover, the baseline has 22% errors and ATT_{ini} has only 9%.

We also counted for each token whether ATT_{ini} was better, equal, or worse than the baseline. The proportions of these cases are shown in Fig. 9.5. Again in Fig. 9.6, for each of the four words, there are far more improvements brought by ATT_{ini} than degradations: on average, 36.2% of the occurrences are improved and only 16.7% are degraded.

Results on WMT datasets.

Moreover, to demonstrate that our findings generalize to larger datasets, we performed an

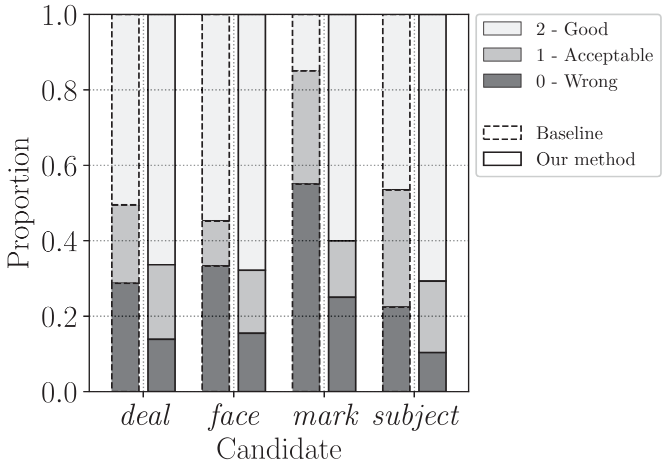


Figure 9.5 – Human comparison of the EN/FR translations of four word types. (a) Proportion of good (light gray), acceptable (middle gray) and wrong (dark gray) translations per word and system (baseline left, ATT_{ini} right, for each word).

experiment on three language pairs from the WMT shared tasks on translation, namely EN-FR, EN-DE and EN-ES. Table 9.4 shows detailed information about the data shared by the WMT organizers.

	Training	Dev	Testing	Labels		Words
				Nouns	Verbs	
EN/FR	5.3M	4.5K	6K	8276	3059	3876
EN/ES	4.5M	3K	5K	7520	1634	3194
EN/NL	3.9M	4.5K	6K	7549	2798	3558

Table 9.4 – Statistics of the WMT corpora used for our additional experiments.

In our previous NMT experiment, we selected 500k parallel sentences for each language pair (from UN for EN/ZH, from Europarl for the other pairs) for training, 5k for development and 50k for testing. The data sets we used for phrase-based MT contain around 2,000 different word forms (after lemmatization) that have more than one sense in WordNet and our WSD system generates around 3.8k different noun labels and 1.5k verb labels on these word forms.

We now experiment with our NMT models using larger datasets over three language pairs, as follows:

(i) Complete EN/DE set from WMT 2016 [Bojar et al., 2016] with a total of ca. 4.5 sentence pairs. The development set is Newstest2013, and the testing sets are Newstest2014 and Newstest2015.

(ii) EN/FR and EN/ES sets are from WMT 2014 [Bojar et al., 2014] with ca. 5.3M sentences for EN/FR and ca. 3.8M sentences for EN/ES. The development sets are Newstest2008 and

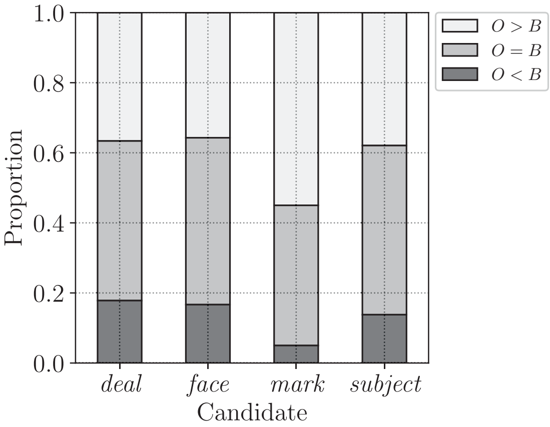


Figure 9.6 – Human comparison of the EN/FR translations of four word types. Proportion of translations in which ATT_{ini} is better (light gray), equal (middle gray) or worse (dark gray) than the baseline.

Newstest2009 while testing sets are Newstest2012 and Newstest2013 for both language pairs.

The larger sets contain ca. 3.5k unique word forms with 8k different noun labels and 2.5k verb labels for each language pair. We use the language sets from different year of WMT because EN/FR and EN/ES data are only available in WMT 2014.

Table 9.5 shows the results of our proposed NMT models on various test sets. The results in this table confirm that our sense-aware NMT models improve significantly the translation quality regardless of the size of the dataset. Moreover, comparing to the results in Table 9.5, our models trained on larger, mixed-domain datasets (WMT) achieve even better improvement than the one trained with smaller (shown in Table 9.2), domain-specific datasets (Europarl Corpus). This clearly shows that our sense-aware NMT models are beneficial on both narrow and broad domains.

	EN/FR		EN/DE		EN/ES	
	NT12	NT13	NT14	NT15	NT12	NT13
Baseline	29.09	29.60	22.79	24.94	32.66	29.57
ATT	29.47 *(+.38)*	30.21 *(+.61)*	23.34 *(+.55)*	25.28 *(+.34)*	33.15 *(+.49)*	30.27 *(+.70)*
ATT_{ini}	**30.26** *(+1.17)*	**30.95** *(+.1.35)*	**23.85** *(+1.06)*	**25.71** *(+.77)*	**34.14** *(+1.48)*	**30.67** *(+1.1)*

Table 9.5 – BLEU scores on *Newstest (NT)* test sets from WMT over three language pairs.

9.3 Comparison to Yang et al. [2017]

Yang et al. [2017] recently proposed to add sense information by using weighted sense embeddings as input to a neural MT system. This is a study that is highly related to ours so we give a comparison between our study and Yang's study in this section.

In Yang et al. [2017], sense labels are generated from a non-parametric skip-gram model [Neelakantan et al., 2014]. For each input token, the latency-controlled context vector is computed from the output of the hidden states of an extra bi-directional RNN and is used to generate the corresponding weight by comparing with each possible sense vector. Finally, the weighted average sense embeddings are directly used as the word embedding for the following encoder.

The differences between our model and Yang et al.'s one are summarized in Table 9.6. First, Yang uses a skip-gram model for sense specification while we use our own adaptive k-means cluster with help of external resource. Moreover, Yang et al. generate the context vectors through an extra bi-directional RNN, while we compute them by averaging the embeddings of all the other words within the current sentence. Finally we represent the word vector by concatenating the embedding of word and the related sense, while Yang et al. use a weighted average of sense embeddings as input to the encoder.

	Yang et al. [2017]	**Ours**
sense label	skip-gram	adaptive k-means
context vector	latency-controlled	aver. neighbour embed.
input embed.	aver. weighted sense embed.	concatenation

Table 9.6 – Difference between our model and Yang et al. [2017]

We conduct an experiment on EN/FR Europarl to compare our models with Yang et al.'s one [Yang et al., 2017]. For that, we reimplemented their model following their description, because their source code is not available. Table 9.7 represents the results of Yang et al.'s model in terms of BLEU using several different settings.

	Yang et al. [2017]		Yang et al. [2017] + convergence	
	random	initialization	random	initialization
EN/FR	30.11	31.05 (-3.55)	33.77 (-0.83)	34.52 (-0.08)
	Baseline 34.60			

Table 9.7 – BLEU scores from Yang et al.'s work with different settings, on EN/FR, using same dataset as our experiment: "initialization" means that the sense embeddings generated by the skip-gram are initialized, while "random" means that embeddings are randomized during preprocessing.

As shown in Table 9.7, using the sense embeddings of multi-sense skip-gram model (MSSG) [Neelakantan et al., 2014] as they do, and training for 6 epochs as in their study, our implementation of their model reaches only 31.05 BLEU points. When increasing the training stage to 15 epochs, the best BLEU score is 34.52, which is still below our NMT baseline of 34.60. We also found that the initialization of embeddings with MSSG brings less than 1 BLEU point improvement with respect to random initializations (which scored 30.11 over 6 epochs and 33.77 over 15 epochs), while Yang et al. found a 1.3–2.7 increase on two different test sets.

By comparing the results in Tables 9.7 and 9.2, we conclude that our models for using sense embeddings outperform Yang et al. [2017]. Based on the comparative results, we draw the following conclusions:

- Our adaptive k-means clustering is better than MSGS for use in NMT;
- In terms of efficiency, Yang et al. need an additional bi-directional RNN to generate the context vector for each input token, while we compute the context vector by averaging the embeddings of the neighboring tokens. Yang et al.'s approach slows down the training of the encoder by a factor of 3, which may explain why they only trained their model for 6 epochs.
- Concatenating the word embedding and its sense vector as input for the RNN encoder is better than just using the sense embedding for each token.

9.4 Conclusion of Part II

We presented both statistical and neural MT systems that leverage sense information, through the use of a larger context than baseline SMT and NMT systems do, through several adaptive context-dependent clustering algorithms for WSD. Our best adaptive clustering method, k-means, provided competitive WSD performance on data from the SemEval 2010 shared task. Then, we integrated the sense information with SMT via a factored model. Furthermore, we presented a NMT model enhanced with an attention-based method to represent multiple word senses. Our experiments with five language pairs showed that both our sense-aware NMT and SMT systems consistently improved over their strong baselines separately, and in particular that they improved specifically the translation of words with multiple senses.

10 Conclusion and Perspectives

In this thesis, we have addressed the issue of translating polysemous words using constraints from lexical consistency and word sense disambiguation that are established at the discourse-level of texts, in wider contexts that can go beyond single sentences. The first part of the thesis was devoted to methods for correcting ambiguous word translations by enforcing translation consistency across sentences, while the second part investigated sense-aware MT systems that address the ambiguity problem for each word.

Conclusion

In the first part of the thesis, we improved the translation of ambiguous words by using lexical consistency. First we presented a method to enforce the consistent translation of a subsequent mention of a compound, when this matches the head noun of the compound. More precisely, we considered XY/Y pairs, hypothesizing that Y should have consistent translations. The occurrences are identified through pattern matching rules, which detect XY compounds followed closely by a potentially co-referent occurrence of Y, such as "Nordwand ... Wand". We proposed one approach to improve the translation of the second occurrence of Y, by post-editing the translation of Y using the head of the translation of XY. We experimented with German-to-French and Chinese-to-English statistical machine translation over the WIT3 and Text+Berg corpora respectively, with 261 XY/Y pairs each. The results showed that baseline SMT systems often translate co-references to compounds consistently for German-to-French, but much less so for Chinese-to-English. For a number of cases in which the noun phrase Y had multiple meanings, our system reduced the frequency of mistranslations in comparison to the baseline, therefore improving noun phrase translation.

Because the XY/Y cases are less frequent than repetitions of nouns Y/Y, we generalized our work by analyzing the repetition of nouns which are not compounds. We presented a method for enforcing consistent translations of repeated nouns, by using machine learning classifiers that are based on lexical, syntactic and semantic features, and decide when consistency should be enforced or not. We first evaluated the accuracy of our classifiers intrinsically, in terms of the accuracy of consistency predictions, on a subset of the UN Corpus. Then, we also evaluated them in combination with phrase-based statistical MT systems on Chinese-English and German-English data. To build our datasets, we detected source-side nouns which appeared twice within a fixed distance and were translated differently by MT. Syntactic features were defined based on the complexity of the parse trees containing the nouns, thus capturing which of the two occurrences of a noun is more syntactically bound, while semantic features focused on the similarity between each translated noun and its context. The trained classifiers were able to predict consistent translations above chance, and, when combined to MT, they bridged 50%-60% of the gap between the baseline MT and an MT system using an oracle classifier.

In the first part of the thesis, we aimed to correct the translation of ambiguous words by either leveraging translation of compound pairs or determining consistency of repeated nouns. In the second part of the thesis, instead of searching for repeated words and post-editing their translations, we designed a sense-aware SMT system that can automatically take the sense knowledge as an additional feature during training and testing. Phrase-based SMT systems use local cues from translation and language models to select the translation of each source word. Such systems do not explicitly perform word sense disambiguation (WSD), although this would enable them to select translations depending on the sense of each word. Previous attempts to constrain word translations based on the results of generic WSD systems have suffered from their limited accuracy. In the second part of the thesis, we demonstrated first a WSD systems that can be adapted to help SMT thanks to three key achievements: we considered a larger context for WSD than SMT could consider; we adapted the number of senses per word to the ones observed in the training data using clustering-based WSD with three clustering algorithms (based on k-means, Chinese restaurant process and random walk); and we initialized sense-clustering with definitions or examples extracted from WordNet.

Our adaptive WSD algorithm leverages semantic information from WordNet to identify the dominant clusters of word occurrences, which correspond to senses, on the source side of a parallel corpus. We found that k-means provided competitive WSD performance on data from the SemEval 2010 shared task. Then, we presented a sense-aware statistical MT system using the decisions from our adaptive WSD as additional features in a factored model. The factored SMT system improved noun and verb translation from English to Chinese, Dutch, French, German and Spanish.

Finally, as the sense integration appeared promising for SMT, we also transferred this approach to the newer neural MT models, which are now state of the art. However, unlike SMT, for which it is easier to use linguistic features, NMT uses word vectors to generate translations, and it is not possible to incorporate sense features as we did for SMT. We designed a neural MT system and compared several methods to use word sense vectors. Our results showed that learning word vectors jointly with sense vectors, which are constructed by our best WSD method, was the best option. We showed that concatenating word and sense vectors, and using a sense selection mechanism based on the weighted average of sense vectors, outperformed several baselines including sense-aware ones. Our experiments with five language pairs showed that our sense-aware NMT system consistently improves over a strong NMT baseline, and in particular it improved specifically the translation of words with multiple senses.

Perspectives

Our study of sense-aware NMT can be extended, as future work, in several ways. First, our results should be confirmed by carrying out further experiments on larger datasets, in particular the bilingual datasets provided for the translation tasks at recent WMT conferences. These time-consuming experiments will enable comparisons with systems participating at WMT, and should improve the training of our system, due to their larger size, at the cost of training time. In fact, in the experiment described in Chapter 9, in order to make a comparison with sense-aware SMT, we used the same datasets (ca. 0.5M words) to train NMT as those used for sense-aware SMT (Chapter 8), although they were smaller than the WMT data sets.

Although we experimented with several language pairs, we always used English as a source language, and therefore only performed word sense disambiguation on one language. We had selected the English language, since the English version of WordNet outperforms other language versions. As a future direction, it is worth exploring semantic resources on other languages (such as German, French and Chinese) and design the adaptive WSD systems that analyse other languages besides English.

In our definition of the ATT_{ini} model in Section 9.1, we computed the context vectors of the current token by averaging the vectors of all the other tokens in the sentence. In future studies, such a computation could be performed across several sentences, by considering the information before and/or after the current sentence. Moreover, a cache-based model can also be considered, which memorizes the previous translations of the tokens. The generated decisions from the cache model can be used as an additional feature during translation, which may lead the NMT system to improve its consistency.

Recently, Vaswani et al. [2017] proposed a new NMT system, the “Transformer”, which is the first sequence transducer model based entirely on attention. This new model achieved state-of-the-art performance on the English-German and English-French translation datasets from WMT. The major innovation of this study is replacing the RNN sequence generation with an attention mechanism, which measures all the tokens of each sentence in the source side parallel and speeds up the training procedure significantly. However, the Transformer does not utilize the sense information from words, and we therefore believe there is still room to improve the performance of the Transformer by integrating the sense information, using the models we proposed in this thesis.

Bibliography

Eneko Agirre and Philip Edmonds. *Word Sense Disambiguation: Algorithms and Applications*. Springer Science & Business Media, 2007.

Eneko Agirre and Aitor Soroa. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41, Athens, Greece, 2009.

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, pages 57–84, 2014.

Andrei Alexandrescu and Katrin Kirchhoff. Graph-based learning for statistical machine translation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 119–127, Boulder, Colorado, 2009.

Eleftherios Avramidis and Philipp Koehn. Enriching morphologically poor languages for statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 763–770, Columbus, Ohio, 2008.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, 2015.

Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145, Berlin, Heidelberg, 2002.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics*, pages 130–138, 2016.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, pages 1137–1155, 2003.

Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. Cache-based online adaptation for machine translation enhanced computer assisted translation. In *MT-Summit*, pages 35–42, 2013.

Alexandra Birch, Miles Osborne, and Philipp Koehn. CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16, Prague, Czech Republic, 2007.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš 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. URL http://www.aclweb.org/anthology/W/W14/W14-3302.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2301.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

Noah Bubenhofer, Martin Volk, David Klaper, Manuela Weibel, and Daniel Wüest. Text+Bergkorpus (release 147_v03), 2013. Digitale Edition des Jahrbuch des SAC 1864-1923, Echo des Alpes 1872-1924 und Die Alpen 1925-2011.

Clara Cabezas and Philip Resnik. Using WSD techniques for lexical selection in statistical machine translation. Technical report, DTIC Document, 2005.

Jimmy Callin, Christian Hardmeier, and Jörg Tiedemann. Part-of-speech driven cross-lingual pronoun prediction with feed-forward neural networks. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 59–64, Lisbon, Portugal, 2015.

Marine Carpuat. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW)*, pages 19–27, Singapore, 2009.

Marine Carpuat and Michel Simard. The trouble with SMT consistency. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 442–449, Korea, 2012.

Marine Carpuat and Dekai Wu. Evaluating the word sense disambiguation performance of statistical machine translation. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005.

Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 61–72, Prague, Czech Republic, 2007.

Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, 2012.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, Prague, Czech Republic, 2007.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar, 2014.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014.

Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. Context-dependent word representation for neural machine translation. *Computer Speech & Language*, 45:149–160, 2017.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, pages 37–46, 1960.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, pages 273–297, 1995.

Wesam Elshamy, Doina Caragea, and William H Hsu. KSU KDD: Word sense induction by clustering in topic space. In *Proceedings of the 5th international workshop on semantic evaluation (SemEval-2010)*, pages 367–370, Los Angeles, California, 2010.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA, USA, 1998.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

William A Gale, Kenneth W Church, and David Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237, USA, 1992.

Zhengxian Gong, Min Zhang, and Guodong Zhou. Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 909–919, Edinburgh, 2011.

S Grin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, pages 107–117, 1998.

Liane Guillou. Improving pronoun translation for statistical machine translation. In *Proceedings of EACL 2012 Student Research Workshop (13th Conference of the European Chapter of the ACL)*, pages 1–10, Avignon, France, 2012.

Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. Findings of the 2016 WMT shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation (WMT16)*, Berlin, Germany, 2016.

Christian Hardmeier. Discourse in statistical machine translation. *Discours*, pages 1–29, 2013.

Christian Hardmeier. *Discourse in Statistical Machine Translation*. PhD thesis, Uppsala University, Sweden, 2014.

Christian Hardmeier and Marcello Federico. Modelling pronominal anaphora in statistical machine translation. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*, Paris, France, 2010a.

Christian Hardmeier and Marcello Federico. Modelling pronominal anaphora in statistical machine translation. In *IWSLT (International Workshop on Spoken Language Translation); Paris, France; December 2nd and 3rd, 2010.*, pages 283–289, 2010b.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL)*, pages 1179–1190, Jeju, Korea, 2012.

Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. Pronoun-focused MT and cross-lingual pronoun prediction: Findings of the 2015 DiscoMT shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16, Lisbon, Portugal, 2015.

Felix Hill, Kyunghyun Cho, Sébastien Jean, and Yoshua Bengio. The representational geometry of word meanings acquired by neural machine translation models. *Machine Translation*, pages 3–18, 2017.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 873–882, Jeju, Korea, 2012.

Shuan Fan Huang. Chinese as a metonymic language. In *In Honor of William Wang: Interdisciplinary studies on Language and Language Change*, pages 223–252, Taipei, Taiwan, 1995.

Masaki Itagaki, Takako Aikawa, and Xiaodong He. Automatic validation of terminology translation consistency with statistical method. *Proceedings of MT summit XI*, pages 269–274, 2007.

David Jurgens and Keith Stevens. Hermit: Flexible clustering for the Semeval-2 WSI task. In *Proceedings of the 5th international workshop on semantic evaluation (SemEval-2010)*, pages 359–362, Los Angeles, California, 2010.

Roman Kern, Markus Muhr, and Michael Granitzer. KCDC: Word sense induction by using grammatical dependencies and sentence phrase structure. In *Proceedings of the 5th international workshop on semantic evaluation (SemEval-2010)*, pages 351–354, Los Angeles, California, 2010.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, pages 79–86, Phuket, Thailand, 2005.

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

Philipp Koehn and Hieu Hoang. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876, Prague, Czech Republic, 2007.

Philipp Koehn et al. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics (ACL), Demonstration Session*, pages 177–180, Prague, Czech Republic, 2007.

Ioannis Korkontzelos and Suresh Manandhar. UoY: graphs of ambiguous vertices for word sense induction and disambiguation. In *Proceedings of the 5th international workshop on semantic evaluation (SemEval-2010)*, pages 355–358, Los Angeles, California, 2010.

Kimmo Koskeniemmi and Mariikka Haapalainen. Gertwol–lingsoft oy. *Linguistische Verifikation: Dokumentation zur Ersten Morpholympics*, pages 121–140, 1994.

Ronan Le Nagard and Philipp Koehn. Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and Metrics (MATR)*, pages 258–267, Uppsala, Sweden, 2010.

Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, New York, USA, 1986.

Jiwei Li and Dan Jurafsky. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal, 2015.

Frederick Liu, Han Lu, and Graham Neubig. Handling homographs in neural machine translation. *arXiv preprint arXiv:1708.06510*, 2017.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *AAAI*, pages 2418–2424, 2015.

Sharid Loaiciga, Thomas Meyer, and Andrei Popescu-Belis. English-French Verb Phrase Alignment in Europarl for Tense Translation Modeling. In *Proceedings of the 9th international conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland, 2014.

Ngoc Quang Luong and Andrei Popescu-Belis. A contextual language model to improve machine translation of pronouns by re-ranking translation hypotheses. In *Proceedings of the 19th Conference of the European Association for Machine Translation (EAMT)*, pages 292–304, Riga, Latvia, 2016.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, pages 2579–2605, 2008.

James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, pages 281–297, Oakland, CA, USA, 1967.

Suresh Manandhar, Ioannis P. Klapaftis, Dmitriy Dligach, and Sameer S. Pradhan. SemEval-2010 task 14: Word sense induction and disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, pages 63–68, Los Angeles, CA, USA, 2010.

Christopher Manning and Dan Klein. Optimization, MaxEnt Models, and Conditional Estimation without Magic. In *Tutorial at HLT-NAACL and 41st ACL conferences*, Edmonton, Canada and Sapporo, Japan, 2003.

Laura Mascarell, Mark Fishel, Natalia Korchagina, and Martin Volk. Enforcing consistent translation of German compound coreferences. In *Proceedings of the 12th Konvens Conference*, pages 58–65, Hildesheim, Germany, 2014.

Thomas Meyer. *Discourse-level Features for Statistical Machine Translation.* PhD thesis, EPFL, Lausanne, 2014.

Thomas Meyer and Andrei Popescu-Belis. Using sense-labeled discourse connectives for statistical machine translation. In *Proceedings of the EACL 2012 Joint Workshop on Exploiting Synergies between IR and MT, and Hybrid Approaches to MT (ESIRMT-HyTra)*, pages 129–138, Avignon, France, 2012.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*, Scottsdale, AZ, USA, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b.

Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41 (2), 2009.

Steven Neale, Luıs Gomes, Eneko Agirre, Oier Lopez de Lacalle, and António Branco. Word sense-aware machine translation: Including senses as contextual features for improved translation models. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2777–2783, Portoroz, Slovenia, 2016.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar, 2014.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167, 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, Philadelphia, PA, USA, 2002.

Ted Pedersen. Duluth-WSI: SenseClusters applied to the sense induction task of SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, pages 363–366, Los Angeles, CA, USA, 2010.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct): 2825–2830, 2011.

Xiao Pu, Laura Mascarell, Andrei Popescu-Belis, Mark Fishel, Ngoc Quang Luong, and Martin Volk. Leveraging compounds to improve noun phrase translation from Chinese and German. In *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop*, pages 8–15, Beijing, China, 2015.

Xiao Pu, Laura Mascarell, and Andrei Popescu-Belis. Consistent translation of repeated nouns using syntactic and semantic cues. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 948–957, Valencia, Spain, 2017a.

Xiao Pu, Nikolaos Pappas, and Andrei Popescu-Belis. Sense-aware statistical machine translation using adaptive context-dependent clustering. In *Proceedings of the Second Conference on Machine Translation*, pages 1–10, Copenhagen, Denmark, 2017b.

J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, 1993.

Alexandre Rafalovitch and Robert Dale. United Nations general assembly resolutions: A six-language parallel corpus. In *Proceedings of MT Summit XII*, pages 292–299, Ontario, Canada, 2009.

Annette Rios, Laura Mascarell, and Rico Sennrich. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark, 2017.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, 2007.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal, 2015.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. SRILM at Sixteen: Update and Outlook. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Waikoloa, Hawaii, 2011.

Jinsong Su, Deyi Xiong, Shujian Huang, Xianpei Han, and Junfeng Yao. Graph-Based collective lexical selection for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1238–1247, Lisbon, Portugal, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.

Jörg Tiedemann. Context adaptation in statistical machine translation using models with exponentially decaying cache. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 8–15, Uppsala, Sweden, 2010.

Ferhan Ture, Douglas W Oard, and Philip Resnik. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 417–426, Korea, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, Long Beach, CA, USA, 2017.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. Word-sense disambiguation for machine translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 771–778, Vancouver, British Columbia, Canada, 2005.

Lesly Miculicich Werlen and Andrei Popescu-Belis. Using coreference links to improve spanish-to-english machine translation. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 30–40, Valencia, Spain, 2017.

Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. Document-level consistency verification in machine translation. In *Proceedings of the 13th Machine Translation Summit*, pages 131–138, Xiamen, China, 2011.

Deyi Xiong and Min Zhang. A sense-based translation model for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1459–1469, Baltimore MD, USA, 2014.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Multi-sense based neural machine translation. In *International Joint Conference on Neural Networks*, pages 3491–3497, Alaska, USA, 2017.

Xiao PU

Rue de Rossettan 40
1920 Martigny
Switzerland
Phone: +41 76 630 15 77
Email: xiao.pu@idiap.ch

Date of birth: Dec. 6, 1987
Nationality: Chinese
Marital status: Married

Summary

Research professional with over five years of combined research and development experience in areas of statistical/neural machine translation, natural language processing (NLP), linguistic feature analysis/extraction, text mining, topic modelling and machine learning. Fast learner, capable of strong analytical and teamwork skills

Education

Aug. 2014 - Present **École Polytechnique Fédérale de Lausanne (EPFL),** Lausanne, Switzerland
Ph.D. in Electrical Engineering
- Advisors: Prof. Dr. Hervé Bourlard and Prof. Dr. Andrei Popescu-Belis

Oct. 2010 - Mar. 2014 **RWTH Aachen University,** Aachen, Germany
M.Sc. in Media Informatics
- GPA: 2.2/6.0[1]
- Thesis: *Interest Mining in Social Networks*

Sep. 2006 - Jul. 2010 **Chongqing University of Post and Telecommunications,** Chongqing, China
B.Sc. in Computer Science
- GPA: 93/100, ranked the top 2nd out of 100

Work Experience

Aug. 2014 - present **Idiap Research Institute,** Martigny, Switzerland
Research Assistant
- Modelling discourse entities and relations for coherent machine translation (MT)
- Study of word sense disambiguation of polysemous words using adaptive context-dependent clustering and its integration in statistical (factored model) and neural (attention mechanism) MT models
- Enhancing statistical MT model for repeated nouns by integration of predictions of several machine learning models learned from semantic and syntactic features

May 2013 - Mar. 2014 **Learning Technologies Research Group, RWTH Aachen University,** Germany
Student Research Assistant
- Parallel analysis of Twitter users text information (AlchemyAPI, OpenCalais, etc)
- Clustering of phrases and keywords by combining outputs from Wikipedia and Latent Dirichlet Allocation (LDA)

Jun. 2011 - Jun. 2012 **Fraunhofer Institute for Production Technology IPT,** Aachen, Germany
Student Research Assistant
- Simulation of manufacturing process of industrial machines such as Abaqus
- Construction of dynamic 2D drawing environment that depends on user input

Skills

Programming:	Python, Java, MATLAB, JavaScript, HTML, SQL, LaTeX
Specific Software:	Moses, PyTorch, Weka, scikit-learn, SciPy
Operating Systems:	Linux, Mac OS X, Windows
Languages:	Chinese (Native), English (fluent), German (B1), French (A0)

Interests

Piano, Guzheng (Chinese zither), traveling, cooking

[1] German grading system with smaller score meaning better performance, score smaller than 2.5 considered as ‘Good’

Journal & Conference:

Xiao Pu, Nikolaos Pappas, James Henderson and Andrei Popescu-Belis (2018) "Integrating Weakly Supervised Word Sense Disambiguation into Neural Machine Translation". In *Transactions of the Association for Computational Linguistics*, 12 pages. (under revision)

Xiao Pu, Nikolaos Pappas and Andrei Popescu-Belis (2017) "Sense-aware Statistical Machine Translation using Adaptive Context-dependent Clustering". In *Proceedings of the Second Conference on Machine Translation (WMT)*, Copenhagen, Denmark, p. 1-10

Xiao Pu, Laura Mascarell and Andrei Popescu-Belis (2017) "Consistent Translation of Repeated Nouns using Syntactic and Semantic Cues". In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Valencia, Spain, p. 948-957

Xiao Pu, Mohamed Amine Chatti and and Ulrik Schroeder (2016) "Wiki-LDA: A Mixed-method Approach for Effective Interest Mining on Twitter Data". In *Proceedings of CSEDU*, Rome, Italy, p. 426-433

Xiao Pu, Laura Mascarell and Andrei Popescu-Belis, Mark Fishel, Ngoc-Quang Luong and Martin Volk. (2015) "Leveraging Compounds to Improve Noun Phrase Translation from Chinese and German". In *Proceedings of the ACL-IJCNLP 2015 Student Session*, Beijing, China, p. 8-15

Xiao Pu and Yang Yong (2010) "Design of a Speech-Recognition Based Internet Search System for the Blind". *Control and Automation*, 2-1: p. 171-173

Talks & Posters:

Andrei Popescu-Belis, **Xiao Pu**, Lesly Miculicich and Laura Mascarell (2017) "Using context to improve the machine translation of nouns and pronouns". Talk at *2nd Swiss Text Analytics Conference (SwissText 2017)*, p. 398-399, Winterthur, Switzerland.

Popescu-Belis A., Evers-Vermeul J., Fishel M., Grisot C., Groen M., Hoek J., Loaiciga S., Luong N.Q., Mascarell L., Meyer T., Miculicich L., Moeschler J., **Pu X.**, Rios A., Sanders T., Volk M. and Zufferey S. (2017) "MODERN: Modeling Discourse Entities and Relations for Coherent Machine Translation". Poster presented at *19th Annual Conference of the European Association for Machine Translation (EAMT)*, Riga, Latvia.