Leveraging Translations for Word Sense Disambiguation

by

Yixing Luan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Word sense disambiguation (WSD) is one of the core tasks in natural language processing and its objective is to identify the sense of a content word (nouns, verbs, adjectives, and adverbs) in context, given a predefined sense inventory. Although WSD is a monolingual task, it has been conjectured that multilingual information, e.g., translations, can be helpful. However, existing WSD systems rarely consider multilingual information, and no effective method has been proposed for improving WSD with machine translation. In this thesis, we propose methods of leveraging translations from multiple languages as a constraint to boost the accuracy of existing WSD systems. Since it is necessary to identify word-level translations from translated sentences, we also develop a novel knowledge-based word alignment algorithm, which outperforms an existing word alignment tool in our intrinsic and extrinsic evaluations. Since our approach is language-independent, we perform WSD experiments on standard benchmark datasets representing several languages. The results demonstrate that our methods can consistently improve the performance of various WSD systems, and obtain state-of-the-art results in both English and multilingual WSD.

Preface

The work presented in this thesis is an extended version of a research article (Luan et al., 2020), which is currently under review. The author of this thesis is the main contributor, who implemented the methods and conducted the experiments.

Acknowledgements

I am grateful to my supervisor, Professor Greg Kondrak for his support and guidance throughout the thesis. I would also like to acknowledge Professor Lili Mou for his assistance with computational resources and research advice.

I would also like to thank Bradley Hauer for the suggestions and feedback.

Finally, I am deeply oblige to my parents for providing me with unfailing support and continuous encouragement. This accomplishment would not have been possible without them.

This thesis is completed with funding from Japan Student Services Organization.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Natural languages are ambiguous in the way that many words have multiple word senses. Usually, word senses can be determined by context. For example, given the noun *bank* in the context shown in Figure 1.1, humans do not have difficulty with disambiguating the sense of "sloping land" and the sense of "financial institution" in each context. However, it becomes a difficult task for computers to disambiguate word senses.

$bank_n^1$: sloping land	clay that Argiento brought from the **bank** of the Tiber
$bank_n^2$: financial institution	appointed by the **bank** administering the estate

Figure 1.1: Examples of different senses for *bank* in different contexts. The examples are from SemCor (Miller et al., 1994), a manually sense-annotated corpus in English.

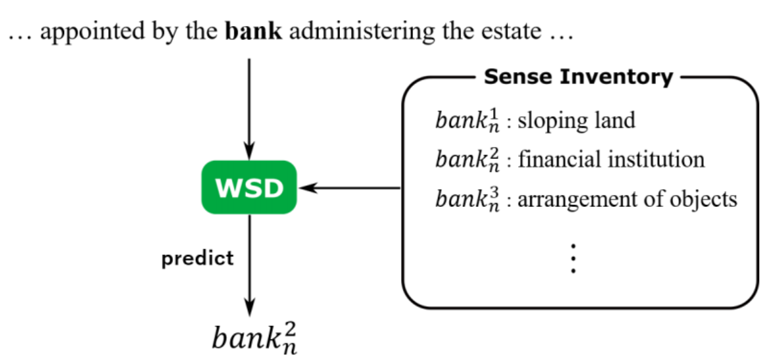

Figure 1.2: Simplified illustration of the WSD task.

Word sense disambiguation (WSD) is the task of identifying the sense of a content word (nouns, verbs, adjectives, and adverbs) in context, given a predefined sense inventory, which enumerates possible word senses for each content word. Thus, WSD can be viewed as a classification task in which the input is a content word in context and the label is a predefined sense, as shown in Figure 1.2. WSD is one of the core tasks in natural language processing with various applications as described in the following examples (Navigli, 2009).

- *Machine Translation*: Machine translation is the task of automatically identifying the target translation for a given source text. Ambiguous words can affect the translation quality because the same source word can have completely different target translations depending on their word senses. For example, when the English noun *bank* is translated into French, the sense of "sloping land" will be translated to *rive*, but the sense of "financial institution" will be translated to *banque*. Thus, disambiguating the word sense beforehand will be beneficial.

- *Information Retrieval*: Information retrieval is the task of obtaining relevant information resources from a given query. Existing search engines usually do not rule out irrelevant web documents containing the query words used in different senses. It becomes possible to prune unrelated documents and increase the search precision by disambiguating the query words and the queried documents.

Also, WSD itself is a meaningful task. For example, it can be used as the assistance for dictionary look-up to help language learners. When language learners look up an ambiguous word in a dictionary, they will find a list of possible senses. However, it is sometimes challenging to identify the correct sense from the context in an unfamiliar language. By applying WSD, we can automate the sense identification and facilitate the language learning process.

A predefined sense inventory is necessary to perform WSD. WordNet (Miller, 1995) is the most widely used sense inventory for English WSD, and it currently covers over 150k English words with over 200k senses. WordNet can also be used as a semantic network showing semantic relations among synsets, the sets of words sharing the same sense (sets of synonyms). Synsets are linked to each other based on semantic relations such as hypernym-hyponym relations. Such information is useful for WSD and many WSD systems take advantage of it (Moro et al., 2014; Agirre et al., 2018).

Although WSD is a monolingual task, it has been conjectured that multilingual information could be helpful (Dagan et al., 1991; Resnik and Yarowsky, 1999; Carpuat, 2009). Attempts have been made to develop methods leveraging parallel corpora for sense tagging (Diab and Resnik, 2002), but no effective method for improving WSD with translations has been proposed to date.

Much of the history of WSD has been determined by the availability of manually created lexical resources in English, including SemCor, a manually sense-annotated corpus, and WordNet, a semantic network. The situation changed with the introduction of BabelNet (Navigli and Ponzetto, 2012a), a massive multilingual semantic network, created by automatically integrating WordNet, Wikipedia, and other resources. BabelNet covers over 250 languages, and in particular, BabelNet synsets contain sets of translations in multiple languages for each individual word sense. Thus, we can view BabelNet synsets as mappings between senses and translations (sense-translation mappings). In Figure 1.3, we show two BabelNet synsets corresponding to the sense of "sloping land" and the sense of "financial institution" respectively for the word *bank*. Methods have been proposed to use multilingual information in BabelNet for WSD (Navigli and Ponzetto, 2012b; Apidianaki and Gong, 2015), but they do not directly exploit the mapping between senses and translations in multiple languages.

While there have been many attempts to apply WSD to machine translation (MT), (Liu et al., 2018; Pu et al., 2018), our goal instead is to harness advances in MT to improve WSD. Rather than develop a new WSD system, we propose general methods that can make existing and future systems more accurate by leveraging translations. We evaluate our methods with several supervised and knowledge-based WSD systems.

Our principal method constrains sense predictions of a given base WSD system using sense-translation mappings from BabelNet. The approach is

$bank_n^1$ **(Synset Id: bn:00008363n)**

Definitions (Gloss)

- WordNet: Sloping land (especially the slope beside a body of water)
- Wikipedia: In geography, the word bank generally refers to the land alongside a body of water.

Examples

- WordNet: They pulled the canoe up on the bank
- WordNet: He sat on the bank of the river and watched the currents

Translations

- EN: bank, beach, coast, riverbank, riverside, shore, ...
- FR: berge, rive, Berges, Chemin de berge, ...
- IT: riva, argine, sponda, banchina, ripa, ...
- JA: 岸, バンク, 川岸, 土手, ...

⋮

$bank_n^2$ **(Synset Id: bn:00008364n)**

Definitions (Gloss)

- WordNet: A financial institution that accepts deposits and channels the money into lending activities
- Wikipedia: A bank is a financial institution that accepts deposits from the public and creates credit.

Examples

- WordNet: He cashed a check at the bank
- WordNet: That bank holds the mortgage on my home

Translations

- EN: bank, depository financial institution, banking company, ...
- FR: banque, Établissement bancaire, Société bancaire, ...
- IT: banca, istituto di credito, banco, cassa, ...
- JA:銀行, 電子決済取引金融機関, バンキング, ...

⋮

Figure 1.3: BabelNet synsets of two different senses for the noun *bank*. (For simplicity, not all information is shown.)

robust enough to take advantage of translations in multiple languages, which are produced manually or by MT models. It is also able to leverage sense frequency information, which can be obtained in either a supervised or an unsupervised manner. To incorporate a more recent technique, we test another method that integrates translations as contextual word embeddings into a WSD system to bias its sense predictions. To obtain word-level translations from the translated contexts, we also introduce a novel alignment algorithm guided by BabelNet synsets.

Our experimental results demonstrate that translations can significantly improve existing WSD systems. We perform several experiments on English and multilingual WSD with both manual and MT translations. In the English WSD experiments with manual translations and word-level alignments, we determine the potential of our methods in an ideal situation. In the multilingual WSD experiments, we demonstrate the language-independence of our methods. Finally, in the English WSD experiments with MT translations, we validate its robustness and effectiveness by showing improvements over existing WSD systems.

The main statement of this thesis is the following: *although WSD is a monolingual task, the performance of existing English and multilingual WSD systems can be improved by leveraging translations from multiple languages as a constraint.*

The main contributions of this thesis are as follows: (1) we propose the first effective method to improve WSD with automatically generated translations; (2) we achieve state-of-the-art results with our language-independent knowledge-based method in both English all-words and multilingual WSD; (3) we introduce an effective bitext alignment algorithm that leverages information from BabelNet.

This thesis is organized as follows. In Chapter 2, we first review various existing WSD systems, and then, introduce prior attempts to improve WSD systems by integrating translations. In Chapter 3, we propose our baseline method HardConstraint and our principal method SoftConstraint, followed by a novel knowledge-based bitext alignment algorithm. Chapter 4 shows intrinsic and extrinsic evaluations to compare our alignment algorithm with an existing alignment tool. In Chapter 5, we test our methods of improving WSD systems with translations in several experimental settings. Finally, Chapter 6 concludes this thesis and discusses future work. We also provide the detailed hyperparameter settings in Appendix A.

Chapter 2

Related Work

This chapter provides a general overview of WSD and translations in prior work. First, we review existing WSD systems. Then, after introducing how WSD is used to improve MT systems, we describe the prior attempts to integrate translations into WSD.

2.1 WSD Systems

There are two main approaches to WSD: supervised and knowledge-based. Supervised systems are trained on sense-annotated corpora and generally outperform knowledge-based systems. On the other hand, knowledge-based systems usually rely only on a semantic network by utilizing graph-based algorithms. Since it is expensive to manually obtain sense-annotated corpora and such corpora exist mainly in English, it is often impractical to apply supervised systems to the multilingual setting. Therefore, for multilingual WSD, knowledge-based approaches are typically employed.

Many effective WSD systems have been proposed. To perform WSD in English, supervised systems are usually trained on SemCor (Miller et al., 1994), a manually sense-annotated corpus in English. IMS (Zhong and Ng, 2010) is a canonical supervised WSD system that trains support vector machines on SemCor to produce word expert models, which provide different models for each word type, with various lexical features such as surrounding words, part-of-speech (POS) tags of surrounding words, and local collocations. Iacobacci et al. (2016) extend IMS by introducing static word embeddings as an additional feature.

Neural approaches are also employed to build supervised systems. Raganato et al. (2017b) propose a bidirectional LSTM model that can produce a unified model to disambiguate all test words to show improvements over classical word expert models. Kumar et al. (2019) propose an extended WSD framework incorporating sense embeddings (EWISE) to address insufficient sense coverage in the training data, i.e., SemCor. Instead of training a model to produce discrete sense labels, EWISE uses a bidirectional LSTM model with self-attention to predict sense embeddings from the test context. EWISE makes sense predictions by comparing the similarity among obtained sense embeddings and the gold sense embeddings, which are derived as knowledge-graph embeddings computed from WordNet.

Nowadays, pre-trained deep models and contextualized word embeddings are shown to be effective for various NLP tasks (Peters et al., 2018; Devlin et al., 2019). LMMS (Loureiro and Jorge, 2019) leverages contextual word embeddings computed by the BERT pre-trained model (Devlin et al., 2019), surpassing the long-standing 70% F-score ceiling for supervised WSD. It learns supervised sense embeddings by applying BERT to SemCor, with additional semantic knowledge from WordNet. LMMS can perform WSD by a 1-nearest neighbor (1-NN) approach. For a given target word, its contextual embedding is also computed through BERT, and it is compared against LMMS embeddings of the possible sense candidates for the target word. Accordingly, the sense of the LMMS embedding that is closest to the target contextual embedding is used as a prediction.

Instead of using contextual embeddings from BERT, Huang et al. (2019) finetune the BERT pre-trained model by adding a classification layer on top of it. To obtain better WSD performance, they concatenate test sentences and WordNet glosses of the possible senses as inputs.

Vial et al. (2019) propose an ensemble of transformer models taking BERT embeddings as inputs. Their sense vocabulary compression (SVC) system achieves state-of-the-art results on English all-words WSD by complementing the sense coverage in the training data, i.e., SemCor, with hypernym-hyponym relations in WordNet.

Among the knowledge-based systems, the Lesk algorithm (Lesk, 1986) is a classic system that determines word senses based on the word overlaps among sense glosses and the context in which the test word appears. Banerjee and Pedersen (2003) extended the Lesk algorithm by additionally considering the related sense glosses based on hierarchical relations in WordNet. More recent knowledge-based systems usually apply graph-based algorithms. Babelfy (Moro et al., 2014) applies random walks with restarts to BabelNet to perform WSD and entity linking, the task of linking entity mentions in context to proper entries in a semantic network. Even though Babelfy is based on BabelNet, it does not utilize the translation information in BabelNet. Similarly, UKB (Agirre et al., 2014, 2018) uses personalized PageRank on WordNet and achieves state-of-the-art performance on English all-words WSD among knowledge-based systems.

Multilingual WSD can be achieved either by automatically developing sense-annotated corpora in multiple languages for training supervised systems or by applying a knowledge-based system. Scarlini et al. (2019) map Wikipedia categories to senses to automatically create sense-annotated corpora OneSeC in multiple languages. When used to train an existing supervised WSD system, it even outperforms the same system trained on SemCor, a manually sense-annotated corpus, in terms of F-score evaluated on English all-words WSD. It also outperforms existing automatic corpora when tested on other languages.

As a multilingual knowledge-based system, SensEmBERT (Scarlini et al., 2020) learns knowledge-based multilingual sense embeddings obtained by combining contextual representations learned using BERT with knowledge obtained from BabelNet. SensEmBERT also employs a 1-NN approach to perform WSD, and it yields state-of-the-art results on English nouns WSD and multilingual WSD.

2.2 WSD for MT

There is some work studying the potential of WSD when integrated into MT systems. Even for recent NMT systems, WSD is also beneficial because existing NMT systems sometimes have difficulties with properly translating ambiguous words despite their ability to encode global sentential context (Rios Gonzales et al., 2017).

Liu et al. (2018) provide empirical evidence showing that translating highly ambiguous words (homonyms) is still challenging for strong NMT systems by showing the translation accuracy on English words with 15 senses (defined by Cambridge English dictionary) is on average 30% lower than the accuracy on monosemous English words, which have only one sense. Also, they propose an NMT system that incorporates context-aware word embeddings to differentiate word senses, and their system improves the quality of translations in terms of both the BLEU score and translation accuracy on ambiguous words.

Pu et al. (2018) also address the issues with translating ambiguous words by proposing a sense-aware NMT system. They employ clustering-based WSD algorithms to induce sense embeddings, which represent probable senses for each source word. By concatenating the learned sense embeddings with the source word embeddings as inputs, they bias the NMT system to properly translate ambiguous words. Their sense-aware NMT system shows consistent improvement over the base NMT system on 5 language pairs.

In this work, we proceed in the reverse direction: we leverage advances in NMT systems to improve the performance of WSD systems.

2.3 Translations for WSD

The integration of multilingual information to improve WSD has been considered in prior work. Through analyzing a multilingual dictionary on small word samples, Resnik and Yarowsky (1999) observe that highly distinct senses can translate differently, and thus can restrict possible sense candidates. However, they do not propose an actual method to perform WSD with translations based on their observation.

Diab and Resnik (2002) propose a WSD system based on translation information extracted from a bitext. They perform WSD based on sense similarities among English words sharing the same translation. Thus, translations are only used to cluster similar English words. In their experiments, they attempt to obtain translations using commercial MT systems, but they did not address the noise introduced by the MT systems. Also, their method fails to outperform systems that rely on monolingual information only. The underlying assumption of their method is that words sharing the same translation are synonymous. However, there is another possibility: such a translation is polysemous. As shown by Yao et al. (2012), these two contradicting assumptions can be both true with almost the same probability. Thus, it is questionable to always assume one of them is correct, and actually, Diab and Resnik (2002) find highly polysemous or homonymous translations hurt the performance of their method.

Cross-lingual WSD (Lefever et al., 2010) is a related task that aims to predict a set of translations for a given ambiguous English word in context. In this task, instead of using a predefined sense inventory, word senses are described by a set of translations in different languages. Apidianaki (2009) use bitexts to create bilingual sense inventory on word samples for cross-lingual WSD. Also, there are some attempts to integrate translations as bag-of-words feature vectors to enhance cross-lingual WSD (Lefever et al., 2011; Lefever and Hoste, 2014). Since the goal of cross-lingual WSD differs from standard WSD, our approach is not directly comparable.

There is also some work leveraging translations available in BabelNet. Navigli and Ponzetto (2012b) make use of translations in BabelNet synsets as a feature in a graph-based WSD system. They follow the recurring idea that translations can restrict possible senses (Dagan et al., 1991; Resnik and Yarowsky, 1999). However, instead of translating the context of the test word, they take into account all the translations of each sense of the test word in BabelNet. Through their English WSD experiment, they show introducing the information from multiple languages yields better performance than the same graph-based system with monolingual information only. Although they use translations to enhance the sense distinctions, they do not explicitly apply translations as a constraint.

Apidianaki and Gong (2015) directly apply sense-translation mappings in BabelNet as a hard constraint on sense predictions using translations from sense-annotated parallel datasets. Unlike this thesis, their approach is applied to the BabelNet First Sense (BFS) baseline, derived from the degree of BabelNet synsets, rather than to an actual WSD system. Also, they only use translations from a single language and do not develop a method that is able to simultaneously integrate translations from multiple languages. In addition, they apply an off-the-shelf word alignment tool only to the test data, which comprises less than 500 sentences, to obtain translations for the test words. Since the accuracy of cooccurrence-based alignment algorithms will be seriously degenerated by the limited size of data, their approach contains many alignment errors. Due to these issues, their results on English WSD fail to show improvement over the simple baseline. Also, when tested in other languages, their method fails to outperform other systems that are dependent on monolingual information only. Furthermore, since their method is proposed for SemEval-2015 task 13 multilingual WSD (Moro and Navigli, 2015), the evaluation on the standard WSD datasets is not performed, as manual translations do not exist.

Chapter 3

Methodology

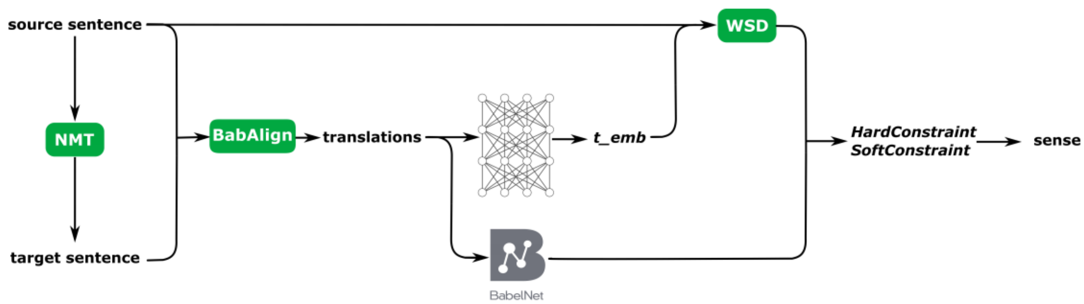

Figure 3.1: The entire architecture of our model.

In our WSD formulation, the input is a sentence, with a word, e, designated as the *focus word*. We are also provided with the set of possible senses of the focus word $S(e)$ from the sense inventory. The task is to determine which sense $s \in S(e)$ is the sense of e in this sentence. We assume that a WSD system assigns some numerical value or score (e.g. probabilities) to each sense, with the output being the sense with the maximum score.

In this chapter, we propose two methods, called HardConstraint and SoftConstraint, which can be used to augment a WSD system that meets our WSD formulation (referred to as a "base" system). Both methods leverage translations for WSD in order to constrain sense predictions made by a base WSD system. In addition, we introduce *t_emb*, a method of leveraging contextual word embeddings to enhance the integration of translations in combination with those constraints. Finally, since our methods crucially depend upon identifying the translation of the focus word in the translated sentence, we also introduce BabAlign, a new knowledge-based word alignment algorithm to further improve the WSD performance. Figure 3.1 shows the entire architecture of our model based on those components.

3.1 HardConstraint

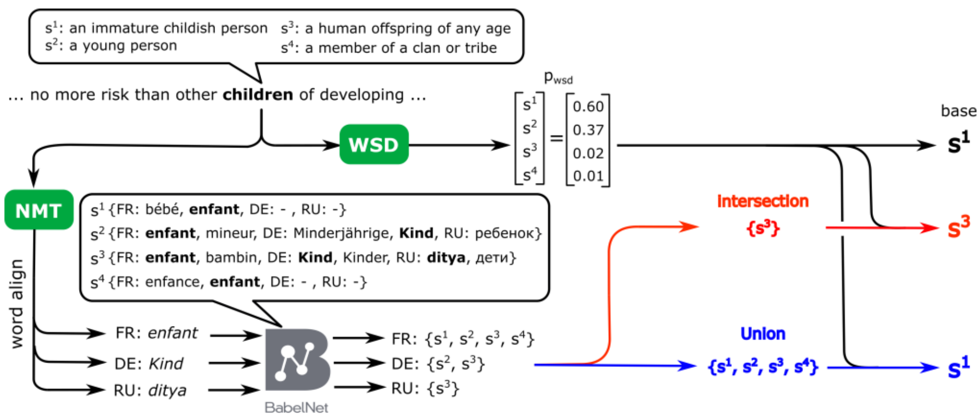

Figure 3.2: The application of HardConstraint with intersection (red) and union (blue) strategies when disambiguating the word *children* in the given context (actual example from Senseval2 data where the correct sense is s^2).

Our first method HardConstraint extends the idea of Apidianaki and Gong (2015) to constrain the set of possible senses of the focus word, i.e., $S(e)$, based on sense-translation mappings in BabelNet. However, instead of relying on a single translation, we incorporate multiple languages through intersection and union strategies.

In the intersection strategy, we take the intersection of the individual sets of senses; that is, we rule out senses if their corresponding BabelNet synsets do not contain translations from all target languages. The intersection strategy is simple but inflexible: the correct sense can be accidentally ruled out if the provided translation of the focus word is not found in the corresponding BabelNet synset. The procedure for making the final sense prediction with HardConstraint (intersection) is shown in Algorithm 1.

On the other hand, in the union strategy, we take the union of the individ-

Algorithm 1 HardConstraint (intersection)

Input:

Set of sense candidates for the source focus word e, $S(e) = \{s_1, \ldots, s_n\}$
Set of target translations of e in different languages, $T(e) = \{t_{L1}, \ldots, t_{Lm}\}$

($\triangleright$ indicates a comment)

1: $\triangleright$ get a list of sense candidates ranked by assigned probabilities
2: $S_{ranked} \leftarrow runWSD(S(e))$
3: $\triangleright$ take the intersection of the individual sets of senses corresponding to BabelNet synsets containing e and $t_L \in T(e)$
4: $S_e^t \leftarrow S(e)$
5: **for** $t_L \in T(e)$ **do**
6:     $S_e^t \leftarrow S_e^t \cap BabelSynsets(e, t_L)$
7: **if** $S_e^t \neq \emptyset$ **then**
8:     **for** s in S_{ranked} **do**
9:         **if** $s \in S_e^t$ **then**
10:             **return** s
11: **else**
12:     **return** $S_{ranked}[0]$

Subroutines:

13: $runWSD(S(e))$ returns the list of sense candidates ranked by assigned probabilities derived from a base WSD system.

14: $BabelSynsets(e, t_L)$ returns the set of senses corresponding to BabelNet synsets containing both the source word e and the target translation t_L.

ual sets of senses; that is, we rule out senses if their corresponding BabelNet synsets do not contain any target translations. This baseline method can somewhat address the inflexibility of the intersection strategy, but it is not as good as the intersection at reducing the number of sense candidates. The procedure for making the final sense prediction with HardConstraint (union) can be shown by changing lines 4 and 6 in Algorithm 1. Instead of getting the whole sense candidates $S(e)$, S_e^t gets $\emptyset$ as an initial state. Also, instead of taking the intersection of the individual sets of senses, S_e^t is updated by taking the union: $S_e^t \leftarrow S_e^t \cup BabelSynsets(e, t_L)$.

Our implementations of HardConstraint consider the intersection or union of the sets of senses corresponding to synsets that contain translations from each language. Ideally, the resulting intersection or union contains exactly one sense, which we take as the final prediction. Otherwise, if they contain multiple senses, we choose the one with the highest score from the base WSD system. If they happen to be empty, we also back-off to the prediction of the base WSD system. In Figure 3.2, we exemplify the entire procedure of HardConstraint with both strategies.

3.2 SoftConstraint

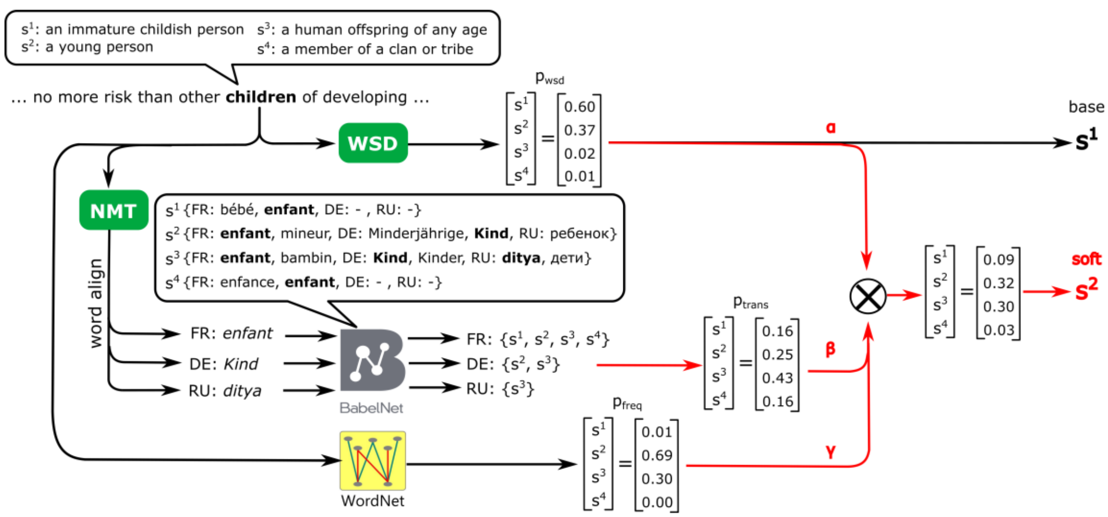

Figure 3.3: The application of SoftConstraint (red) when disambiguating the word *children* in the given context (actual example from Senseval2 data where the correct sense is s^2).

HardConstraint is effective at ruling out sense candidates, but also quite sensitive to MT errors and BabelNet deficiencies. BabelNet contains translations for only 79% of the nominal senses in WordNet, and its multilingual lexicalizations have an average precision of only 72% (Navigli and Ponzetto, 2012a).

Our principal method, SoftConstraint, is more robust in handling noisy MT translations and BabelNet gaps. It integrates information from three sources: the base WSD system, translations, and sense frequencies (Figure 3.3). From each of these sources, we derive a probability distribution over $S(e)$. We employ the product of experts (PoE) approach (Hinton, 2002) to combine the probabilities as follows:

$$\tilde{p}(s) = p_{wsd}(s)^{\alpha} \cdot p_{trans}(s)^{\beta} \cdot p_{freq}(s)^{\gamma}$$

The resulting score $\tilde{p}$ is an unnormalized measure of probability with tunable weights α, β, and γ, which sum up to one. We tune those weights through grid-search on held-out development sets. The sense that maximizes this measure is taken as the prediction. Below, we provide the details on each of the three distributions.

Probability p_{wsd} is obtained by simply normalizing the numerical scores from the base WSD system.

Probability p_{trans} is calculated on the basis of the set of translations for each source focus word e in BabelNet. Given a source focus word e and a word f in another language, we obtain its sense coverage $c(e, f)$ representing the number of possible senses of e that are mapped to f, i.e., the number of BabelNet synsets containing both e and f. Based on the sense coverage, the word pair e and f is assigned a weight $w(e, f)$ that reflects its discrimination power:

$$w(e, f) = \begin{cases} \frac{1}{c(e,f)} & \text{if } c(e, f) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Now, we consider f to be a translation $t_L(e)$ for e in a target language $L \in \mathcal{L}$, where $\mathcal{L}$ stands for the set of target languages. The score of a candidate sense $s \in S(e)$ is then the sum of weights of the translations that are found in the corresponding BabelNet synset $BN(s)$:

$$score(s) = \sum_{L \in \mathcal{L}} \{\mathbb{1}_{BN(s)}(t_L(e)) \cdot w(e, t_L(e))\}$$

where $\mathbb{1}_{BN(s)}(t_L(e))$ is an indicator function that becomes 1 if $t_L(e) \in BN(s)$ and 0 otherwise. As with p_{wsd}, we normalize the scores into a proper probability distribution p_{trans} over the set of senses. Also, to avoid zero values, we perform smoothing by adding a small positive value (a tunable parameter). For example, p_{trans} of each sense for *children* in Figure 3.3 can be computed as follows:[1]

$$w(\textit{FR: enfant}) = \frac{1}{|\{s^1, s^2, s^3, s^4\}|} = \frac{1}{4} = 0.25$$
$$w(\textit{DE: Kind}) = \frac{1}{|\{s^2, s^3\}|} = \frac{1}{2} = 0.5$$
$$w(\textit{RU: ditya}) = \frac{1}{|\{s^3\}|} = \frac{1}{1} = 1.0$$

$$score(s^1) = w(\textit{enfant}) = 0.25$$
$$score(s^2) = w(\textit{enfant}) + w(\textit{Kind}) = 0.75$$
$$score(s^3) = w(\textit{enfant}) + w(\textit{Kind}) + w(\textit{ditya}) = 1.75$$
$$score(s^4) = 0.0$$

$$p_{trans}(s^1) = \frac{score(s^1)}{\sum_{s \in S} score(s)} = \frac{0.25}{1.75} \simeq 0.14$$
$$p_{trans}(s^2) = \frac{score(s^2)}{\sum_{s \in S} score(s)} = \frac{0.5}{1.75} \simeq 0.29$$
$$p_{trans}(s^3) = \frac{score(s^3)}{\sum_{s \in S} score(s)} = \frac{1.0}{1.75} \simeq 0.57$$
$$p_{trans}(s^4) = \frac{score(s^4)}{\sum_{s \in S} score(s)} = 0.0$$

Probability p_{freq} represents the sense frequency information for a given lemma and POS. This information is also used by most WSD systems. For English, we obtain sense frequencies from WordNet, which derives such information from SemCor, a sense-annotated corpus. To handle senses with zero frequency in SemCor, we also apply additive smoothing. To obtain p_{freq} for languages other than English, which lack large, high-quality sense-annotated corpora, we use CluBERT (Pasini et al., 2020), the state-of-the-art system for unsupervised sense distribution learning, which applies a clustering algorithm to contextual embeddings from BERT (Devlin et al., 2019) to infer the

[1] Resulting scores are slightly different from scores in Figure 3.3 because smoothing weight is omitted for simplicity.

frequency distribution of the senses of a given word from an un-annotated corpus. Like our methods, CluBERT is language-independent, has no additional training data requirements, and has been successfully integrated into WSD systems to improve their performance.

Figure 3.3 illustrates how SoftConstraint combines the three probability distributions to repair an incorrect sense prediction produced by a base system.

3.3 Contextual Word Embeddings

Recent work has demonstrated the utility of contextual word embeddings for NLP tasks (Peters et al., 2018; Devlin et al., 2019). Accordingly, WSD systems such as SensEmBERT (Scarlini et al., 2020) take a contextual embedding of the focus word as input, in order to leverage its dense encoding of relevant local information, which may be used to determine the correct sense.

In this section, we propose a method of adding translation information to the input of a WSD system by modifying the contextual embedding of the focus word to reflect its translation. We refer to this method as *t_emb*. Note that this method can be combined with either the HardConstraint or SoftConstraint methods. Unlike those methods, which use translations of the focus word to post-process the output of a WSD system, *t_emb* provides the translation information in the form of a contextual embedding directly as input to the WSD system. Thus, translation information is used as an additional feature to bias sense predictions of the base WSD system.

As before, our approach is to translate the context of the focus word, and use word alignment to identify the translation of the focus word. We compute a contextual embedding of this translation, just as we did for the focus word itself, and then concatenate the two embeddings. This produces a new embedding that can be provided to a base WSD system in place of the focus word embedding alone. However, since not all WSD systems use contextual embeddings, this method is less general, and we only apply it in some of our models and evaluation experiments.

3.4 Translation Alignment

The effectiveness of our approach to improving WSD depends on the correct identification of the word-level translations in each language. Even when the sentential context of the focus word is correctly rendered in another language, both HardConstraint and SoftConstraint rely on the proper alignment between the source focus word and its translation, which may be composed of multiple word tokens. Although attention weights in some NMT systems may be used to derive word alignment, such an approach is not necessarily more accurate than off-the-shelf alignment tools (Li et al., 2019). Therefore, our approach is to instead identify the word-level translations by performing a bitext-based alignment between the source focus words and their translations.

During development, we found that the accuracy of alignment tools such as FastAlign (Dyer et al., 2013) is limited by the size of the aligned bitext, as well as the lack of access to the translation information that is present in BabelNet. To mitigate these issues, we introduce a knowledge-based word alignment algorithm BabAlign[2] that leverages translation information in BabelNet by post-processing the output of an off-the-shelf word aligner. Starting from the test sentences in our WSD data and their translations, we first append the translated WSD data to a large lemmatized bitext to ensure enough amount of the input data for the aligner. We further augment the input data with the BabelNet translations for all WSD focus words to bias the aligner to predict the alignment links to valid translations, sharing BabelNet synsets with the source focus words. We then run the base aligner in both translation directions, and take the intersection of the two sets of alignment links.

In its final stage, BabAlign leverages the BabelNet translation pairs again, to post-process the generated alignment. We accept without further correction all alignment links that align a focus word to a content word, which appears in BabelNet as one of valid translations for the focus word. Otherwise, we attempt to find a correct alignment for the focus word by searching for one of its BabelNet translations within the target sentence (*babelex_search*). If

[2]Implementation is available at https://github.com/YixingLuan/BabAlign

a possible translation is composed of multiple words (e.g., French translation *salle d'audience* for English source word *courtroom*), we attempt to expand a partial alignment to a complete alignment by searching the adjacent word tokens until we reach a token aligned to another source token or a function word token (*compound_search*). Thus, our alignment algorithm is strongly guided by its objective of identifying all BabelNet synsets that contain the focus word and its translation. Algorithm 2 shows the entire procedure in BabAlign.

Note that BabAlign assumes one-to-one alignment from the base aligner. If the base aligner produces many-to-many alignment, BabAlign takes the leftmost alignment link as the prediction of the base aligner. Even though BabAlign restricts one-to-one alignment as its input, it can restore many-to-many alignment through its functionality to search surrounding words to detect tokenized compounds.

Algorithm 2 BabAlign

Input:

list of all source tokens in a given sentence, $\sigma_s = (w_{s1}, \ldots, w_{sl})$
list of all target tokens in the translated sentence, $\sigma_t = (w_{t1}, \ldots, w_{tm})$
BabelNet translations for a source word, $Babelex(w_s) = \{lex_1, \ldots, lex_n\}$

($\triangleright$ indicates a comment)

$\triangleright$ assume perfect tokenization in the source side and treat source compounds in σ_s as one token
$A \leftarrow BaseAligner(\sigma_s, \sigma_t)$
$\triangleright$ A is a set of alignment pairs (w_s, w_t) produced by the base aligner;
$\triangleright$ if w_s is not aligned, $w_t = None$

for each $(w_s, w_t) \in A$ **do**
  if $w_t \in Babelex(w_s)$ **then**
    $\triangleright$ search the surrounding words to recover the compound
    $w_t \leftarrow compound\_search(w_s, w_t)$
  else
    $\triangleright$ search the sentence to find a possible BabelNet translation
    $lex \leftarrow babelex\_search(w_s)$
    if $lex \neq None$ **then**
      $w_t \leftarrow lex$
$\triangleright$ return a set of (source focus word, aligned translation) pairs
return A

Subroutines:

$compound\_search(w_s, w_t)$ returns the longest sequence of tokens $lex \in \sigma_t$
such that $lex \in Babelex(w_s)$
and lex contains w_t
and lex does not contain any target tokens (except w_t) that are aligned by the base aligner

$babelex\_search(w_s)$ returns the longest sequence of tokens $lex \in \sigma_t$
such that $lex \in Babelex(w_s)$
and lex does not contain any tokens that are already aligned
$babelex\_search(w_s)$ returns $None$ if no such lex can be found

Chapter 4

Word Alignment Evaluation

To show the effectiveness of BabAlign, which combines an existing word aligner with translations from BabelNet, we evaluate the alignment performance through both intrinsic and extrinsic evaluation.

4.1 Intrinsic Evaluation

To perform an intrinsic evaluation, we use parallel datasets with gold alignment to directly evaluate the alignment performance. After describing the experimental setup, we provide the results and error analysis.

4.1.1 Experimental setup

We employ FastAlign as the base aligner. As the evaluation datasets, we use SemCor 3.0[1] and its translations, Multi SemCor (MSC) (Bentivogli and Pianta, 2005) and Japanese SemCor (JSC) (Bond et al., 2012), to evaluate English-Italian and English-Japanese alignment respectively. Both MSC and JSC contain manually annotated gold alignment for a subset of the sense-annotated content words in SemCor. We extract all English, Italian, and Japanese sentence triples where an English token has gold alignments in both the Italian and Japanese sides. We get 639 sentence triples with 2,602 aligned tokens. We only evaluate the alignment performance for those 2602 sense-annotated tokens, and do not consider the alignment for other tokens, because

[1] We use SemCor 3.0 in the Natural Language Toolkit (NLTK) to keep the compatible file format with MSC and JSC.

our purpose here is to obtain proper translations for test words in the WSD setting.

We experiment in two evaluation settings. For the source side, i.e., SemCor, we continue to use the annotated tokenization, lemma, and POS information in both settings. For the target side, i.e., MSC or JSC, in one setting, we do not use the tokenization, lemma, and POS information provided in the data, and instead, we employ morphological taggers to perform pre-processing: TreeTagger (Schmid, 1994) for Italian, and MeCab (Kudo, 2005) for Japanese. In the other setting, we also use annotated tokenization, lemma, and POS information for MSC and JSC. The former (*un-annotated*) emulates the setting where we generate translations for monolingual WSD datasets, and the latter (*annotated*) shows the alignment performance in the ideal situation. The additional bitexts we append to the data are the OpenSubtitles2018 English-Italian (37.8M sentences) and English-Japanese (2.2M sentences) bitexts (Lison and Tiedemann, 2016). Those bitexts are also pre-processed by morphological taggers in both settings (We also use TreeTagger for the English side of bitexts.) We compute F-score to evaluate alignment performance in terms of whether the lemma of the aligned translation corresponds to the lemma of the manually aligned translation in MSC or JSC.

4.1.2 Results

Table 4.1 compares the alignment approaches. As expected, the concatenation of a large bitext to the test data (+OpenSub) dramatically reduces the number of errors. The addition of translation pairs from BabelNet (+pairs) yields further gains. This shows that our idea of biasing the aligner with BabelNet translations is effective to improve alignment quality. BabAlign substantially improves the quality of the alignment on English-Japanese by nearly 10 points. The improvement on English-Italian is smaller, as the alignment between similar languages is easier, and the additional bitext is much larger. Japanese is particularly challenging, not only because it is typologically different, but also due to the frequency of multi-character compounds. In addition, in the *annotated* setting where morphological information exists in both source and

Method	Data	En-It		En-Ja	
		un-annotated	annotated	un-annotated	annotated
FastAlign	test data only	80.4	85.5	36.0	39.9
	+OpenSub	93.3	96.4	75.6	79.8
	+OpenSub +pairs	93.6	97.2	81.9	90.9
BabAlign	+OpenSub +pairs	**94.0**	**97.9**	**91.6**	**95.7**

Table 4.1: Alignment F-score (%) on English-Italian and English-Japanese bitexts.

target sides, alignment quality increases, and BabAlign shows very accurate alignment.

The back-off strategy used by BabAlign effectively leverages possible translations in BabelNet to recover tokenized compounds and missing alignment links. This mitigates the effect of alignment errors on our WSD results, which we describe in the next chapter.

4.1.3 Error Analysis

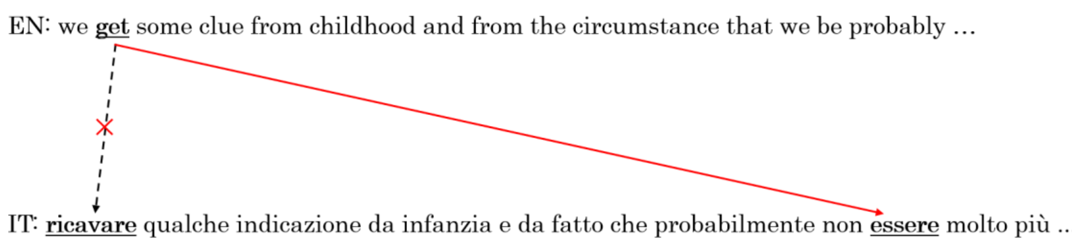

Figure 4.1: The alignment error caused by BabelNet deficiency.

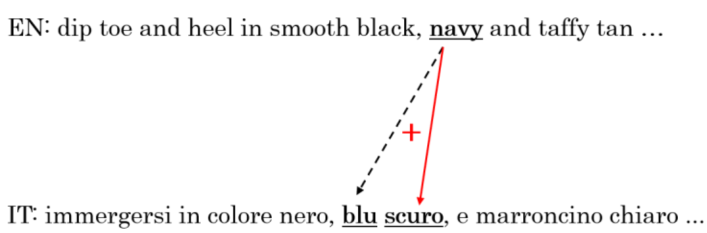

Figure 4.2: The alignment error caused by a tokenization error in MSC.

As shown in Table 4.1, BabAlign is very accurate. For example, in the *annotated* setting of English-Italian, BabAlign gets alignment links for 2,557 instances with 33 errors out of 2,602 instances in total. Most of those 33 errors are originally from FastAlign and could not be fixed by BabAlign either because the translation is not covered by BabelNet or because a proper translation happens to be aligned to another source token sharing the BabelNet synset with it.

There are only six instances where BabAlign hurts the correct alignment link made by FastAlign, and they are caused by two types of errors: one is due to the deficiency in BabelNet (*type1*), and the other is due to the tokenization errors in the dataset (*type2*).

Figure 4.1 shows an example of the *type1* error. Although FastAlign properly aligns the source word *get* to the target translation *ricavare*, BabAlign denies this alignment link because *get* and *ricavare* never occur in the same BabelNet synset. Also, a similar Italian word *essere*, which shares BabelNet synsets with *get*, happens to appear in the same sentence. Since *essere* is not aligned to any source word by FastAlign, BabAlign wrongly takes it as a new alignment link.

An example of the *type2* error is shown in Figure 4.2. Although FastAlign aligns *navy* to *blu* ("blue"), BabAlign properly expands the alignment link to *blu scuro* ("dark blue") to get a more accurate translation. However, in MSC, *blu scuro* is tokenized into two separate tokens, and only *blu* is aligned to *navy*. Thus, the new alignment link made by BabAlign is improperly determined as a wrong alignment link.

The *type2* error indicates the potential use of BabAlign for tokenization error correction in a given data. In our English-Italian test set with given tokenization, BabAlign expands alignment links for three instances to obtain compounds through *compound_search* function, and two of them are correct translations, showing wrong tokenizations in MSC dataset. Thus, it could be possible to develop a tokenization error correction algorithm based on *compound_search* function in BabAlign.

4.2 Extrinsic Evaluation

To perform an extrinsic evaluation, we apply BabAlign to cross-lingual lexical entailment (LE). Closs-lingual LE is the task introduced by Vyas and Carpuat (2016), and they define this task as "the task of detecting whether the meaning of a word in one language can be inferred from the meaning of a word in another language".

In the following evaluation, we perform cross-lingual binary LE, which treats cross-lingual LE as a binary classification task. Thus, given a pair of words in different languages, it aims to detect if one word entails the other. For example, if the given word pair is (EN: *plant*, IT: *rosa*), the answer will be either the word pair holds the entailment relation (positive) or does not hold the entailment relation (negative). In this example, the answer is positive because the Italian word *rosa* ("rose") entails the English word *plant*.

4.2.1 Experimental setup

We again employ FastAlign as the base aligner. To perform cross-lingual LE, we perform word alignment on bitexts to extract lexical translation pairs, based on the assumption that a word and its aligned translation either represents the same concept or one entails the other (Hauer et al., 2020b). Thus, if the test word pair exists in the extracted translation pairs, we determine the test word pair holds the entailment relation. In the example of (EN: *plant*, IT: *rosa*), we determine this word pair holds the entailment relation if *plant* and *rosa* are aligned in the English-Italian bitext.

As the test datasets, we use German-English, German-Croatian, German-Italian, and English-Italian test sets from SemEval-2020 Task 2: Predicting Multilingual and Cross-Lingual Lexical Entailment (Glavaš et al., 2020). Each test set contains around 2,000 to 3,000 word pairs. We use OpenSubtitles bitexts for all language pairs, and the statistics of each bitext are shown in Table 4.2. To perform lemmatization and POS tagging, we employ Reldi-Tagger (Ljubesic et al., 2016) for Croatian and TreeTagger for other languages.

Languages	de-en	de-hr	de-it	en-it
lines	22.5M	13.8M	13.6M	35.2M
bytes	2.7G	1.0G	1.1G	2.6G

Table 4.2: The bitext size for each language pair.

4.2.2 Results

Method	data	de-en	de-hr	de-it	en-it	Average
FastAlign	OpenSub	31.2	32.6	26.3	60.2	37.6
BabAlign	OpenSub +pairs	**52.4**	**41.5**	**40.9**	**61.5**	**49.1**

Table 4.3: F-score (%) on cross-lingual binary lexical entailment test sets.

As can be seen in Table 4.3, BabAlign yields substantial improvements over the base aligner FastAlign in all language pairs. These results can be interpreted as clear evidence that the accurate word alignment produced by BabAlign is highly beneficial for downstream tasks.

BabAlign contributes to the cross-lingual LE performance by detecting more alignment links that hold entailment relations. Since word pairs often show hypernym-hyponym relations when one entails the other, such word pairs do not always share synsets in BabelNet. However, there are still some word pairs sharing BabelNet synsets even though they hold entailment relations. For example, the Italian word *lavoro* ("labor") entails the English word *employment*, and these two words share a BabelNet synset. Thus, for such word pairs, BabAlign can leverage translations in BabelNet to detect the alignment links. Also, even though many BabelNet translation pairs added to the bitexts do not hold entailment relations, they are still useful to improve the alignment accuracy on other content words, which are not in question. This results in narrowing down the choice of alignment links for the test words and improving the overall alignment accuracy.

In addition, BabAlign also improves the cross-lingual LE performance by denying false positives produced by FastAlign. Sometimes, FastAlign happens to align test word pairs that do not show entailments. Since such word pairs barely share BabelNet synsets, BabAlign can avoid those false positives based on BabelNet translations. For example, the English word *river* and the Italian word *signore* ("man") are unrelated to each other. However, in our English-Italian bitext, FastAlign improperly aligns those two words, and thus, produces a false positive. On the other hand, since *river* and *signore* obviously do not share a BabelNet synset, BabAlign can deny the alignment link produced by FastAlign and avoid such a false positive.

Chapter 5

WSD evaluation

In this section, after describing how we replicate the base WSD systems that we use in our experiments, we show how our methods can improve existing WSD systems in the oracle setting for English all-words WSD. Then, we report the results of the experiments on multilingual WSD with both manual and automatic translations. In the end, we evaluate our methods on English all-words WSD with automatic translations.

5.1 WSD System Replication

In the following experiments, we employ various knowledge-based and supervised WSD systems to test how our methods can improve the base systems. Before applying our methods to base systems, we compute probability distributions p_{wsd} from base systems and ensure we can replicate reported results from obtained p_{wsd} by choosing the sense assigned the highest probability. We show replication results for all base systems in Tables 5.1 and 5.2.

Among knowledge-based systems, Babelfy (Moro et al., 2014) is provided as an API[1] with the functionality of outputting p_{wsd} instead of just showing the resulting sense predictions. Babelfy has a variant that take advantage of WordNet first sense (WN1st sense), the sense ranked first in WordNet based on its sense frequency. Moro et al. (2014) set a fixed confidence threshold as 0.8 for WN1st sense back-off. Our replication results are very close (-0.6% F-score on the concatenation of all test datasets) to the reported results in

[1] http://babelfy.org/

System		SE-2	SE-3	SE-07	SE-13	SE-15	ALL
Babelfy + WN1st	reported	67.0	63.5	51.6	66.4	70.3	65.5
	ours	66.6	65.5	53.0	63.0	68.5	64.9
UKB + dict_weight	reported	68.8	66.1	53.0	68.8	70.3	67.3
	ours	68.8	66.1	53.0	68.8	70.3	67.3
IMS	reported	70.9	69.3	61.3	65.3	69.5	68.4
	ours	71.3	69.1	61.5	65.1	68.3	68.3
LMMS	reported	76.3	75.6	68.1	75.1	77.0	75.4
	ours	76.3	75.4	67.9	75.0	76.9	75.3
SVC	reported	79.7	77.8	73.4	78.7	82.6	79.0
	ours	79.7	77.8	73.4	78.7	82.6	79.0

Table 5.1: Replication results on English all-words WSD datasets.

	SE-13				SE-15	
	DE	ES	FR	IT	ES	IT
reported	78.0	74.6	78.0	69.6	64.1	66.0
ours	77.3	74.8	78.5	70.4	64.7	67.7

Table 5.2: Replication results of SensEmBERT on multilingual datasets.

Raganato et al. (2017a), which shows the performance of several WSD systems on standard benchmark datasets. The difference is perhaps due to the absence of the information about the detailed parameter settings.

Agirre et al. (2014, 2018) provide a UKB package[2] with the best-performing parameter settings reported in Agirre et al. (2018), which shows state-of-the-art results on English all-words WSD among knowledge-based systems. UKB has a variant that uses complete sense frequency distributions in WordNet, which are referred to as the dictionary weight (*dict_weight*). Using the provided package, we can obtain p_{wsd} and get the same F-score as Agirre et al. (2018).

As a state-of-the-art multilingual knowledge-based system, Scarlini et al. (2020) provide SensEmBERT sense embeddings in 5 languages.[3] Therefore, following Scarlini et al. (2020), we employ the multilingual BERT cased pre-trained model (768 embedding dimension)[4] made available by Devlin et al. (2019) to compute test word embeddings for WSD based on a 1-nearest neigh-

[2]https://ixa2.si.ehu.es/ukb/
[3]http://sensembert.org/
[4]https://github.com/google-research/bert

bor (1-NN) approach. We take the sum of embeddings from the top 4 layers. Also, when WordPiece tokenization in BERT splits one token into several sub-tokens, we take the average of embeddings for all sub-tokens. We observe our replication results are very close (+0.4% F-score on average) to Scarlini et al. (2020).

We use IMS (Zhong and Ng, 2010) for both English and multilingual WSD experiments. Zhong and Ng (2010) provide a Java package for IMS with built-in English-specific lemmatizer and POS-tagger.[5] In the multilingual WSD experiments, those built-in pre-processors are disabled. Since IMS requires XML files with a particular structure as inputs, we convert training and test datasets before running IMS. We use default parameters defined in the given package. For English WSD, we replicate the results on the standard benchmark datasets reported in Raganato et al. (2017a), and we obtain almost the same results (-0.1% F-score on the concatenation of all test datasets). Note that the original probability distributions produced by IMS do not cover all senses because SemCor does not contain training instances for all WordNet senses. Thus, when computing p_{wsd}, we add small smoothing to the missing senses, which originally get zero probabilities, to fully take advantage of available sense-translation relations. This does not change the results of the base system because we ensure the added probabilities are much smaller than probabilities of other senses that appear during the training.

As a recent supervised system, we use LMMS (Loureiro and Jorge, 2019) for English WSD. Loureiro and Jorge (2019) provide both the pre-trained LMMS sense embeddings and the source code to train LMMS embeddings.[6] We take the pre-trained sense embeddings to replicate the reported results. To obtain test word embeddings for 1-NN based WSD, we employ BERT large cased pre-trained model (1024 embedding dimension). Following Loureiro and Jorge (2019), we also take the sum of the top 4 layers and take the average of all sub-tokens. As a result, we obtain almost the same results (-0.1% F-score on the concatenation of all test datasets) with Loureiro and Jorge (2019).

[5]https://www.comp.nus.edu.sg/ nlp/software.html

[6]https://github.com/danlou/LMMS

As a state-of-the-art supervised system, Vial et al. (2019) provide the source code of their SVC system for replication.[7] They also provide the model checkpoints for their best-performing ensemble models, and thus, we can obtain exactly the same numbers reported in Vial et al. (2019). However, their source code does not store its sense predictions but only shows the resulting F-score. Therefore, we modify the source code to store p_{wsd} and ensure we can also obtain the same F-score by the sense predictions derived from the stored p_{wsd}.

5.2 Oracle WSD Experiments

Our first set of experiments aims at estimating the upper limits of our approach in an oracle setting of annotated and aligned bitexts with high-quality human translations.

5.2.1 Experimental Setup

As described in Section 4.1, our sense-annotated bitexts are MSC and JSC, which contain manual translations of texts from SemCor. As in Section 4.1, we use 639 sentences with 2602 sense-annotated instances, which have manually aligned translations in both MSC and JSC. We randomly sample 10% of the instances as the development set. We tune all parameters on the development set, and use the same hyperparameters throughout the experiment.

We employ two knowledge-based WSD systems: Babelfy and UKB. Since existing supervised systems are usually trained on SemCor, our test set, we do not employ supervised systems in this set of experiments. As mentioned in Section 5.1, both systems have variants that take advantage of sense frequency information in WordNet. Babelfy backs off to WN1st sense using a fixed confidence threshold, which we set to 0.8 following Moro et al. (2014). UKB uses complete sense frequency distributions (*dict_weight*). We use the same hyperparameter settings as Agirre et al. (2018). For a fair comparison, when applying SoftConstraint to a system variant without sense frequency

[7] https://github.com/getalp/disambiguate

System	base	hard(intersect)	hard(union)	soft
Babelfy	50.7	66.7	60.1	**68.6**
UKB	58.0	72.2	64.4	**73.3**
Babelfy + WN1st	72.6	73.4	73.0	**73.6**
UKB + dict_weight	71.2	77.8	74.4	**80.1**

Table 5.3: WSD F-score (%) on SemCor test set with Italian and Japanese translations.

System	Translation	base	hard	soft
Babelfy	IT	50.7	60.3	58.6
	JA		65.8	65.8
UKB	IT	58.0	64.1	64.2
	JA		72.0	72.1
Babelfy + WN1st	IT	72.6	73.2	73.6
	JA		73.1	73.6
UKB + dict_weight	IT	71.2	73.6	75.4
	JA		78.5	80.0

Table 5.4: WSD F-score (%) on SemCor test set with translations from only a single language.

information, we set our γ to 0 to turn off the p_{freq} component.

5.2.2 Results

The results in Table 5.3 demonstrate the effectiveness of leveraging translations for WSD. The systems without sense frequency information are boosted by 15-18%, while the systems with full features get up to 9% absolute improvement. Also, SoftConstraint consistently outperforms HardConstraint. The modest improvement on Babelfy with WN1st sense is due to the base system falling back on WN1st sense in about 77% of test instances, precluding the use of translations.

In additional ablation experiments shown in Table 5.4, we observe that our approach is effective in combining translations from multiple languages. For instance, the F-score of 73.3% for plain UKB with SoftConstraint (shown in Table 5.3) drops to 72.1% with only Japanese translations, and to 64.2% with only Italian translations, vs. 58.0% with no translations. These results also indicate that translations from a more distant language, i.e., Japanese, work better at discriminating senses. We hypothesize the reason is that they share fewer senses with the source words than translations from a close language, i.e., Italian. The verification of this hypothesis is left for future work.

5.3 Multilingual WSD Experiments

Since our methods are language-independent, we test our methods on standard multilingual WSD datasets.

5.3.1 Experimental Setup

We perform our multilingual WSD evaluation on benchmark parallel datasets in English, Spanish, Italian, French, and German from SemEval-2013 task 12 (Navigli et al., 2013) and SemEval-2015 task 13 (Moro and Navigli, 2015).[8] The datasets contain manual reference translations, but are not word-aligned. In our experiments, we only test on languages other than English, and English is always the target side used to obtain translations. We perform experiments in two settings, with either machine or human translations. To obtain automatic translations, we translate the test sets into English using Google Translate (GT)[9] because the pre-trained NMT models for test languages are not always available. For manual translations, we use the provided parallel datasets in all languages. For instance, when we test on the Italian test set in SemEval2013, we use the English, French, Spanish, and German test sets to obtain target translations. For each individual language, we use BabAlign to obtain translations of the focus word in other languages. We randomly sample 10% of test instances in each dataset to obtain development sets for parameter tuning.

We use two multilingual base WSD systems: IMS (Zhong and Ng, 2010) and SensEmBERT (Scarlini et al., 2020). We train IMS on OneSeC (Scarlini et al., 2019), an automatically sense-annotated set of corpora in multiple

[8] French and German are in SemEval-2013 only.
[9] https://translate.google.com/

languages.[10] For SensEmBERT embeddings, when we integrate the translation embedding (t_emb), we concatenate the focus word embedding and its corresponding t_emb, as described in Section 3.3. To compute these contextual word embeddings for English translations[11], we use the 768-dimensional multilingual BERT cased pre-trained model (mBERT). Since both OneSeC and SensEmBERT are limited to nouns, we follow Scarlini et al. (2019, 2020) in performing the evaluation on nominal instances only.

Since languages other than English lack large sense-annotated corpora, we employ two evaluation settings. In the default setting, sense frequency information is not used, with the parameter γ set to 0 in SoftConstraint. In the other setting, we approximate sense distributions with CluBERT (Pasini et al., 2020).

5.3.2 Results

In Tables 5.5 and 5.6, we report the WSD results on SemEval-2013 and SemEval-2015 datasets when applying our methods to IMS and SensEmBERT. Our methods show up to 10% improvement over the state-of-the-art system SensEmBERT, and such a substantial gain can be seen with IMS as well. Surprisingly, the results with English translations from GT are only slightly lower on average than with manual translations from multiple languages, which shows that our methods work well with both types of translations. HardConstraint performs well in this set of experiments, as nouns are very well represented in BabelNet.[12] Hence, HardConstraint barely rules out gold senses and is able to reduce the number of sense candidates without hurting them. For similar reasons, SoftConstraint often gets the zero smoothing weight[13] when computing p_{trans} and results in the same

[10] Iacobacci et al. (2016) propose an extended version of IMS that incorporates static English word embeddings; however, we are not aware of any IMS version with contextual word embeddings.

[11] Even when human translations for multiple languages are available, we only use English translations for t_emb to avoid noise when combining multiple embeddings.

[12] Over 99% of the words in BabelNet are nouns (Navigli and Ponzetto, 2012a). On average, we found 92% of the SemEval translations are in the BabelNet synsets of the correct senses.

[13] Detailed parameter settings are shown in Appendix A.

	Method	SE-13 DE	ES	FR	IT	SE-15 ES	IT	Average
	base system	72.7	67.8	69.6	68.1	63.0	64.1	67.6
GT	hard	73.8	70.6	71.2	74.7	64.6	71.3	71.0
	soft($\gamma = 0$)	73.7	71.4	73.3	74.9	65.0	70.8	71.5
	soft(CluBERT)	72.4	76.8	73.9	75.5	68.2	75.7	73.8
Manual	hard(intersect)	72.0	71.2	74.3	73.4	65.5	70.0	71.1
	hard(union)	73.4	68.8	70.8	73.2	63.5	69.8	69.9
	soft($\gamma = 0$)	73.5	75.0	**74.6**	**76.2**	65.5	71.1	72.7
	soft(CluBERT)	**73.8**	**77.0**	74.5	74.9	**69.1**	**76.5**	**74.3**

Table 5.5: WSD F-score (%) of IMS (OneSeC) with translations on the nominal instances of the SemEval-2013 and SemEval-2015 datasets.

	Method	SE-13 DE	ES	FR	IT	SE-15 ES	IT	Average
	base system	76.7	74.7	77.6	70.7	64.4	68.7	72.1
GT	hard	77.7	80.8	79.4	76.8	64.2	74.1	75.5
	soft($\gamma = 0$)	77.7	80.8	79.4	76.8	65.0	74.1	75.6
	soft(CluBERT)	78.1	80.4	80.7	78.9	65.7	**78.7**	77.1
	soft(CluBERT+t_emb)	78.2	80.8	80.9	79.4	65.9	**78.7**	77.3
Manual	hard(intersect)	77.1	80.1	79.3	76.6	63.5	72.8	74.9
	hard(union)	76.5	78.1	78.9	74.8	64.6	72.5	74.2
	soft($\gamma = 0$)	76.8	**81.9**	80.8	78.3	64.6	73.6	76.0
	soft(CluBERT)	76.8	79.2	**81.5**	**79.8**	66.4	**78.7**	77.1
	soft(CluBERT+t_emb)	**79.6**	81.4	**81.5**	78.9	**66.6**	**78.7**	**77.8**

Table 5.6: WSD F-score (%) of SensEmBERT with translations on the nominal instances of the SemEval-2013 and SemEval-2015 datasets.

performance with HardConstraint when using translations from a single language.

SoftConstraint achieves an average improvement of several F1 points on both systems, even without sense frequency information. The best results are obtained with SoftConstraint using sense frequencies from CluBERT, especially when they can be combined with mBERT-based contextual translation embeddings (*t_emb*), neither of which requires manually sense-annotated corpora. We observe that using *t_emb* is beneficial especially when the translation constraints can only show a small improvement, e.g., SemEval-2013 German. When much noise appears in translations and BabelNet, the efficacy of the translation constraints will degenerate, but *t_emb* can effectively capture

	IMS (OneSec)			SensEmBERT		
Method	Manual	GT	NMT	Manual	GT	NMT
base system		72.7			76.7	
hard	73.3	73.8	73.7	77.2	77.7	77.6
soft($\gamma = 0$)	73.5	73.7	74.0	77.2	77.7	77.6
soft(CluBERT)	73.0	72.4	72.8	77.5	78.1	78.1
soft(CluBERT+t_emb)	-	-	-	78.9	78.2	79.2

Table 5.7: WSD F-score (%) of IMS (OneSeC) and SensEmBERT on the nominal instances of the SemEval-2013 German dataset when using manual English translations and automatic English translations from Google Translate and the NMT model. (CluBERT+t_emb is not applicable with IMS.)

translated contextual information.

We consider our comparison is fair because we do not employ any additional resources that require manual efforts. Since both OneSec and SensEmBERT are based on BabelNet, the only resource we additionally leverage is translation information either from the provided test data or from a publicly available MT model. Thus, we interpret these results as the new state of the art in multilingual WSD based on the consistent improvement over the current state-of-the-art knowledge-based system SensEmBERT.

To evaluate the potential of using translations from a replicable NMT model, we perform an additional experiment. We obtain English translations for test words in the SemEval-2013 German dataset with a pre-trained transformer model for German-English (Ng et al., 2019) available in the fairseq toolkit[14] (Ott et al., 2019). In this setting, as with Google Translate, we only use English as the target language to obtain translations for both constraints and *t_emb*. Table 5.7 shows that the results on both WSD systems with the pre-trained NMT model are almost the same as with Google Translate, and slightly better than with English-only manual translations. According to our preliminary analysis, MT translations may sometimes work better because they tend to be more literal, and easier to correctly align with the source focus words. This suggests that our methods can effectively leverage translations from different kinds of sources.

[14]https://github.com/pytorch/fairseq

5.4 English WSD Experiments with NMT

In the final set of experiments, we evaluate our methods on standard monolingual benchmark datasets using NMT translations from multiple languages.

5.4.1 Experimental Setup

We evaluate on five English all-words datasets: Senseval2, Senseval3, SemEval-2007, SemEval-2013, and SemEval-2015 from the unified framework made available by Raganato et al. (2017a). We test our methods with five base WSD systems. As knowledge-based systems, we employ Babelfy (Moro et al., 2014) and UKB (Agirre et al., 2014, 2018). As supervised systems, we employ IMS (Zhong and Ng, 2010), LMMS (Loureiro and Jorge, 2019), and SVC (Vial et al., 2019), trained on SemCor 3.0 provided in Raganato et al. (2017a). We tune parameters on Senseval2, and apply the same parameter settings in all datasets. We compare plain Babelfy and UKB to SoftConstraint without p_{freq}. For other systems, we derive p_{freq} from sense frequency information available in WordNet 3.0.

Since those test datasets are not accompanied by translations, we automatically obtain the translations from pre-trained transformer-based NMT models available in the fairseq toolkit: English-French and English-German models from Ott et al. (2018), and an English-Russian model from Ng et al. (2019). Note that unlike multilingual WSD experiments (Section 5.3), we do not use Google Translate in the following experiments.

As with the previous experiments, we apply BabAlign to obtain word-level alignment among source focus words in the test dataset and target translations produced by NMT models.

5.4.2 Results

Table 5.8 shows the results on the standard English all-words WSD datasets. While HardConstraint with both strategies is not sufficiently robust to improve complex WSD systems with automatically generated translations, SoftConstraint shows statistically significant improvements over the original performance for all base systems except for SVC. Since SVC is very accurate, it correctly predicts over 75% of the instances, for which we could find at least one BabelNet translation, limiting the benefit from translations.

Also, Table 5.9 shows substantial gains occur on nominal instances because nouns are the major components in BabelNet as mentioned in Section 5.3.2.

In summary, these results again demonstrate that our knowledge-based method can effectively integrate information from the WSD system itself, translations, and sense frequency even with noisy translations generated by NMT models and with noise in BabelNet.[15] While translations are shown to help even strong supervised WSD systems, the improvements are particularly impressive on knowledge-based systems. The SoftConstraint result on UKB with *dict_weight* sets a new state of the art for knowledge-based systems.

5.5 Error Analysis

Compared with HardConstraint, SoftConstraint is more beneficial in two situations. The first situation is that SoftConstraint can fully take advantage of sense-translation mappings from BabelNet to correct the wrong sense prediction by the base system even when HardConstraint cannot. For example, UKB with *dict_weight* cannot predict the sense of "arrangement" for the focus word *order* in the test sentence *"... at a signal, the ringers begin varying the* ***order*** *in which the bells sound ..."*, but predicts the sense of "command" instead. The French translation *ordre* shares the BabelNet synsets with *order* for all 15 senses, and thus, it is not useful for ruling out sense candidates. Also, the Russian translation *porjadok* does not appear in the BabelNet synset for the sense of "arrangement". On the other hand, the German translation *reihenfolge* is only covered by the correct BabelNet synset. Therefore, HardConstraint with intersection cannot find proper intersection including the correct sense, and it fails to correct the prediction by the

[15] Due to the complexity of transforming mBERT representations into different dimensionalities and vector spaces, translation embeddings are not used in these experiments.

	System	Method	SE-2	SE-3	SE-07	SE-13	SE-15	ALL
Knowledge-based	WN1st sense baseline	-	66.8	66.2	55.2	63.0	67.8	65.2
	Babelfy	base system	50.2	46.4	38.9	55.6	54.3	50.3
		hard(intersect)	53.0*	49.2*	41.7*	55.6	55.9*	52.3*
		hard(union)	52.8*	50.7*	43.5*	57.9*	56.3*	53.3*
		soft($\gamma = 0$)	**57.7***	**54.3***	**47.0***	**60.1***	**61.8***	**57.3***
	UKB	base system	64.2	54.8	40.0	**64.5**	64.5	60.4
		hard(intersect)	65.3*	57.4*	44.0*	62.6	66.2*	61.5*
		hard(union)	65.9*	57.8*	42.2	64.4	66.3*	62.1*
		soft($\gamma = 0$)	**67.6***	**58.8***	**48.6***	**64.5**	**71.1***	**64.0***
	Babelfy + WN1st	base system	66.6	65.5	53.0	63.0	**68.5**	64.9
		hard(intersect)	66.7	65.5	53.4	62.7	**68.5**	64.9
		hard(union)	66.9	65.7	53.0	62.9	**68.5**	65.0
		soft	**67.4***	**65.9**	**54.3***	**63.4**	68.3	**65.4***
	UKB + dict_weight	base system	68.8	66.1	53.0	68.8	70.3	67.3
		hard(intersect)	68.5	65.5	53.6	64.5	69.7	66.1
		hard(union)	69.6	66.2	51.9	67.8	71.3	67.4
		soft	**71.3***	**66.8**	**54.1**	**69.0**	**74.2***	**68.9***
Supervised	IMS	base system	71.3	**69.1**	**61.5**	65.1	68.3	68.3
		hard(intersect)	71.0	68.2	60.7	62.0	67.6	67.1
		hard(union)	71.1	67.5	58.5	63.7	68.8	67.4
		soft	**72.3**	68.7	59.8	**65.8**	**71.7***	**69.0***
	LMMS	base system	76.3	75.4	67.9	75.0	76.9	75.3
		hard(intersect)	75.9	74.1	66.2	70.9	75.7	73.6
		hard(union)	76.0	72.3	64.4	72.4	76.5	73.6
		soft	**77.2**	**77.1***	**69.2**	**76.1**	**77.2**	**76.4***
	SVC	base system	79.7	**77.8**	**73.4**	**78.7**	**82.6**	**79.0**
		hard(intersect)	78.2	75.4	71.0	72.9	80.0	76.1
		hard(union)	77.9	74.1	67.9	75.4	80.6	76.1
		soft	**80.1**	77.7	72.7	**78.7**	82.0	**79.0**

Table 5.8: English all-words WSD F-score (%) on standard evaluation datasets with translations from 3 languages (French, German, and Russian). The results show statistically significant improvement over the base system are marked with * (McNemar's Test, $p < 0.05$).

	System	Method	Nouns	Verbs	Adj.	Adv.	All
knowledge-based	WN1st sense	base	67.6	50.3	74.3	80.9	65.2
	Babelfy	base	57.6	32.3	51.2	38.0	50.3
		hard(intersect)	59.3*	35.4*	52.4*	40.9*	52.3*
		hard(union)	60.2*	36.8*	53.2*	41.3	53.3*
		soft	**64.1***	**42.6***	**54.7***	**44.3***	**57.3***
	UKB	base	65.7	39.9	69.3	68.2	60.4
		hard(intersect)	66.2	42.3*	**69.7**	71.4*	61.5*
		hard(union)	67.3*	42.4*	68.6	72.3*	62.1*
		soft	**69.1***	**46.4***	66.8	**76.9***	**64.0***
	Babelfy + WN1st	base	67.3	50.2	74.1	80.1	64.9
		hard(intersect)	67.3	50.2	74.2	80.1	64.9
		hard(union)	67.5	50.2	73.8	**80.9**	65.0
		soft	**67.9***	**50.5**	**74.7**	**80.9**	**65.4***
	UKB + dict_weight	base	71.2	50.7	75.0	77.7	67.3
		hard(intersect)	69.0	50.7	74.9	79.2	66.1
		hard(union)	71.4	51.1	73.3	79.8	67.4
		soft	**72.6***	**52.9***	**75.9**	**80.6**	**68.9***
supervised	IMS	base	70.2	**56.4**	**75.1**	83.5	68.3
		hard(intersect)	68.4	55.7	**75.1**	83.2	67.1
		hard(union)	69.7	54.9	73.0	83.5	67.4
		soft	**71.7***	55.6	74.8	**84.4**	**69.0***
	LMMS	base	77.9	63.8	**80.8**	83.5	75.3
		hard(intersect)	75.4	63.0	80.1	84.7	73.6
		hard(union)	76.8	60.3	78.6	83.5	73.6
		soft	**79.1***	**65.5***	80.2	**85.3**	**76.4***
	SVC	base	81.4	**68.7**	**83.7**	85.5	**79.0**
		hard(intersect)	77.3	66.7	83.1	85.8	76.1
		hard(union)	79.2	63.5	80.8	85.5	76.1
		soft	**81.5**	68.2	**83.7**	**86.4**	**79.0**

Table 5.9: English all-words WSD F-score (%) on each POS in the concatenation of all five datasets with translations from 3 languages (French, German, and Russian). The results show statistically significant improvement over the base system are marked with * (McNemar's Test, $p < 0.05$).

base system. Also, HardConstraint with union fails to reduce the number of sense candidates at all due to the French translation, and thus, it keeps the base system's prediction as is. Unlike HardConstraint, SoftConstraint effectively takes advantage of translations (especially German) and sense frequency information to correctly predict the sense of "arrangement".

The second situation is that SoftConstraint is robust to noise in MT translations and the incompleteness of BabelNet so that it can avoid miscorrecting the proper sense prediction by the base system. For example, UKB with *dict_weight* correctly predicts the sense of "earth" for the focus word *world* in *"... **world**'s two dozen most influential countries ..."*. However, English *world* and its three translations, *monde*, *Welt*, and *mir*, are only found in the BabelNet synset glossed as "populace", while the Russian translation *mir* happens to be missing from the BabelNet synset glossed as "earth" (perhaps because there is no Russian link to the English Wikipedia page for *World*). Hence, while HardConstraint miscorrects the UKB prediction to the sense of "populace", SoftConstraint keeps it unchanged by leveraging sense frequencies and the base system scores.

Although SoftConstraint is more robust, there are still some instances where translations hurt the base system. For example, UKB with *dict_weight* correctly predicts the sense of "energy" for the focus word *zip* in *"... requires **zip** in the way of athletic prowess ..."*. However, the NMT models wrongly translate *zip* in the sense of "fastener" for all languages. Thus, all of the French (*zip*), German (*reißverschluss*), and Russian (*molnija*) translations only appear in the BabelNet synset for the sense of "fastener". Since all translations are wrong, none of our methods can keep the correct prediction by the base system.

Chapter 6

Conclusion

In this thesis, we proposed a novel approach to improving WSD by leveraging translations from multiple languages, which incorporates a novel knowledge-based bitext alignment. Since our methods are not designed for any particular base WSD systems or test languages, we tested them on several systems in both English and multilingual WSD settings. We demonstrated experimentally that SoftConstraint can consistently improve WSD performance even when no manual translations are available, leading to state-of-the-art results on knowledge-based English all-words and multilingual WSD. We also demonstrated our novel alignment algorithm BabAlign can substantially outperform an existing word alignment tool in both intrinsic and extrinsic evaluations. In short, we empirically tested our statement: the performance of existing English and multilingual WSD systems can be improved by leveraging translations. Also, we established our contributions to formulating the methods of leveraging automatic translations and showing the effectiveness of our methods throughout our WSD experiments.

Although our method achieved state-of-the-art results for knowledge-based English all-words and multilingual WSD, there are several directions for further research. Regarding our method of integrating contextual translation embeddings (*t_emb*), we only applied *t_emb* to multilingual WSD experiments due to the complexity of mapping translation embeddings to different embedding spaces. We plan to investigate a more general method to integrate *t_emb* so that we can validate the advantage of *t_emb* in the English all-words

WSD setting as well. Since not all supervised systems are significantly improved by our post-processing constraint methods in English all-words WSD experiments, we expect integrating *t_emb* will be helpful by introducing more abundant information about translations and senses.

Also, it will be interesting to test our methods in other types of tasks related to WSD. For example, Pilehvar and Camacho-Collados (2019) propose word in context (WiC) challenge, a binary classification task of detecting if the same word appearing in the pair of sentences share the same meaning. We plan to apply our methods to this task to validate our methods of leveraging translations can be helpful for not only the standard WSD task but also a more general task that requires disambiguating word meanings.

In addition, we would like to test BabAlign as a tokenization error correction method as described in our intrinsic evaluation. Even in manually constructed corpora such as Multi SemCor, we found a few tokenization errors that are detected and fixed by BabAlign. Thus, applying BabAlign for tokenization error correction will be more beneficial for automatic corpora, which are very important to WSD especially in languages other than English.

References

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2018. The risk of sub-optimal use of open source nlp software: Ukb is inadvertently state-of-the-art in knowledge-based wsd. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 29–33.

Marianna Apidianaki. 2009. Data-driven semantic analysis for multilingual wsd and lexical selection in translation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*, pages 77–85.

Marianna Apidianaki and Li Gong. 2015. LIMSI: Translations as source of indirect supervision for multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 298–302.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, volume 3, pages 805–810.

Luisa Bentivogli and Emanuele Pianta. 2005. Exploiting parallel texts in the creation of multilingual semantically annotated resources: The MultiSemCor Corpus. *Natural Language Engineering*, 11(3):247–261.

Francis Bond, Timothy Baldwin, Richard Fothergill, and Kiyotaka Uchimoto. 2012. Japanese SemCor: A sense-tagged corpus of Japanese. In *Proceedings of the 6th Global WordNet Conference (GWC 2012)*, pages 56–63.

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

Ido Dagan, Alon Itai, and Ulrike Schwall. 1991. Two languages are more informative than one. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 130–137, Berkeley, California, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mona Diab and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 255–262.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Goran Glavaš, Ivan Vulić, Anna Korhonen, and Simone Ponzetto. 2020. SemEval-2020 task 2: Predicting multilingual and cross-lingual (graded) lexical entailment. In *Proceedings of the 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Arnob Mallik, and Grzegorz Kondrak. 2020a. Low-resource G2P and P2G conversion with synthetic training data. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–122, Online. Association for Computational Linguistics.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Arnob Mallik, and Grzegorz Kondrak. 2020b. Ualberta at SemEval-2020 task 2: Using translations to predict cross-lingual entailment. In *Proceedings of the 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Rashed Rubby Riyadh, and Grzegorz Kondrak. 2019a. Cognate projection for low-resource inflection generation. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 6–11, Florence, Italy. Association for Computational Linguistics.

Bradley Hauer, Yixing Luan, and Grzegorz Kondrak. 2019b. You shall know the most frequent sense by the company it keeps. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 208–215. IEEE.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3500–3505.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907.

Taku Kudo. 2005. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.net/*.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Els Lefever and Véronique Hoste. 2014. Parallel corpora make sense: Bypassing the knowledge acquisition bottleneck for word sense disambiguation. *International Journal of Corpus Linguistics*, 19(3):333–367.

Els Lefever, Véronique Hoste, and Martine De Cock. 2010. Semeval-2010 task 3: Cross-lingual word sense disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 15–20, Uppsala, Sweden. Association for Computational Linguistics.

Els Lefever, Véronique Hoste, and Martine De Cock. 2011. ParaSense or how to use parallel corpora for word sense disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 317–322, Portland, Oregon, USA. Association for Computational Linguistics.

Michael Lesk. 1986. 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. ACM.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016*, pages 923–929. European Language Resources Association.

Frederick Liu, Han Lu, and Graham Neubig. 2018. Handling homographs in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1336–1345. Association for Computational Linguistics.

Nikola Ljubesic, Filip Klubicka, Zeljko Agic, and Ivo-Pavao Jazbec. 2016. New inflectional lexicons and training corpora for improved morphosyntactic annotation of croatian and serbian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Yixing Luan, Bradley Hauer, Lili Mou, and Grzegorz Kondrak. 2020. Improving word sense disambiguation with translations. In submission.

George A Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, pages 240–243. Association for Computational Linguistics.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297. Association for Computational Linguistics.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231.

Roberto Navigli and Simone Paolo Ponzetto. 2012a. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli and Simone Paolo Ponzetto. 2012b. Joining forces pays off: Multilingual joint word sense disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1399–1410, Jeju Island, Korea. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

Tommaso Pasini, Federico Scozzafava, and Bianca Scarlini. 2020. CluBERT: A Cluster-Based Approach for Learning Sense Distributions in Multiple Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL-HLT*, pages 1267–1273, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167.

Philip Resnik and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. 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. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2019. Just “OneSeC” for producing multilingual sense-annotated data. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 699–709, Florence, Italy. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New methods in language processing*, pages 44–49.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. *CoRR*, abs/1905.05677.

Yogarshi Vyas and Marine Carpuat. 2016. Sparse bilingual word representations for cross-lingual lexical entailment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1187–1197, San Diego, California. Association for Computational Linguistics.

Xuchen Yao, Benjamin Van Durme, and Chris Callison-Burch. 2012. Expectations of word sense in parallel corpora. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 621–625. Association for Computational Linguistics.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83.

Appendix A

Parameters

System	Translations	δ_{trans}	δ_{freq}	α	β	γ
Babelfy	IT	0.04	-	0.3	0.7	-
	JA	0.01	-	0.1	0.9	-
	IT+JA	0.06	-	0.1	0.9	-
UKB	IT	0.01	-	0.1	0.9	-
	JA	0.01	-	0.1	0.9	-
	IT+JA	0.04	-	0.2	0.8	-
Babelfy + WN1st	IT	0.01	0.01	0.1	0.1	0.8
	JA	0.01	0.01	0.1	0.1	0.8
	IT+JA	0.06	0.01	0.1	0.1	0.8
UKB + dict_weight	IT	0.01	0.01	0.2	0.3	0.5
	JA	0.01	0.01	0.2	0.4	0.4
	IT+JA	0.01	0.01	0.1	0.7	0.2

Table A.1: Tuned parameters used to obtain English all-words WSD results reported in Section 5.2.2.

System	δ_{trans}	δ_{freq}	α	β	γ
Babelfy	0.01	-	0.1	0.9	-
UKB	0.01	-	0.3	0.7	-
Babelfy + WN1st	0.01	0.01	0.1	0.4	0.5
UKB + dict_weight	1.00	0.02	0.1	0.8	0.1
IMS	0.48	0.01	0.5	0.4	0.1
LMMS	0.87	0.01	0.8	0.1	0.1
SVC	0.01	0.01	0.3	0.5	0.2

Table A.2: Tuned parameters used to obtain English all-words WSD results reported in Section 5.4.2.

	Test Language	δ_{trans}	α	β	γ
GT	SE-13 DE	0.00	0.1	0.1	0.8
	SE-13 ES	0.00	0.1	0.1	0.8
	SE-13 FR	0.00	0.8	0.1	0.1
	SE-13 IT	0.00	0.1	0.5	0.4
	SE-15 ES	0.00	0.1	0.1	0.8
	SE-15 IT	0.00	0.1	0.8	0.1
Manual	SE-13 DE	0.01	0.1	0.3	0.6
	SE-13 ES	0.00	0.1	0.5	0.4
	SE-13 FR	0.00	0.5	0.4	0.1
	SE-13 IT	0.00	0.1	0.8	0.1
	SE-15 ES	0.01	0.2	0.3	0.5
	SE-15 IT	0.00	0.1	0.8	0.1

Table A.3: Tuned parameters for the best performing method (SoftConstraint with CluBERT) applied to IMS (OneSeC) to obtain multilingual WSD results reported in Section 5.3.2.

	Test Language	δ_{trans}	α	β	γ
GT	SE-13 DE	0.00	0.8	0.1	0.1
	SE-13 ES	0.00	0.8	0.1	0.1
	SE-13 FR	0.00	0.2	0.3	0.5
	SE-13 IT	0.00	0.1	0.1	0.8
	SE-15 ES	0.01	0.6	0.1	0.3
	SE-15 IT	0.00	0.1	0.1	0.8
Manual	SE-13 DE	0.00	0.1	0.1	0.8
	SE-13 ES	0.00	0.1	0.2	0.7
	SE-13 FR	0.00	0.1	0.7	0.2
	SE-13 IT	0.00	0.4	0.5	0.1
	SE-15 ES	0.02	0.7	0.1	0.2
	SE-15 IT	0.00	0.1	0.1	0.8

Table A.4: Tuned parameters for the best performing method (SoftConstraint with CluBERT and *t_emb*) applied to SensEmBERT to obtain multilingual WSD results reported in Section 5.3.2.