University of Toronto, Department of Computer Science
CSC 2501F—Computational Linguistics, Fall 2023

Reading assignment 3

Due date: Electronically by noon, Friday, 13 October 2023.
Late write-ups will not be accepted without documentation of a medical or other emergency.
This assignment is worth 3% of your final grade.

What to read

1. A. Raganato, J. Camacho-Collados and R. Navigli, "Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison," *Proc. EACL*, 2017, 99–110.

2. W. Lu and T. H. Nguyen, "Similar but not the Same: Word Sense Disambiguation Improves Event Detection via Neural Representation Matching," *Proc. EMNLP*, 2018, 4822–4828.

What to write

Write a *brief* summary of these two papers, with a critical assessment of their merits.

Some points to consider:

- What is transfer learning?
- How should we evaluate word-sense disambiguation?

General requirements: Your write-up should be typed, using 12-point font and 1.5-line spacing; it should fit on one to two sides of a sheet of paper. Hand in one copy at the beginning of class.

Submission is electronic and can be done from any Teaching Labs machine using the `submit` command:

```
$ submit -c csc2501h -a Essay3 essay3.pdf
```

Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison

Alessandro Raganato, Jose Camacho-Collados and **Roberto Navigli**
Department of Computer Science
Sapienza University of Rome
{raganato,collados,navigli}@di.uniroma1.it

Abstract

Word Sense Disambiguation is a long-standing task in Natural Language Processing, lying at the core of human language understanding. However, the evaluation of automatic systems has been problematic, mainly due to the lack of a reliable evaluation framework. In this paper we develop a unified evaluation framework and analyze the performance of various Word Sense Disambiguation systems in a fair setup. The results show that supervised systems clearly outperform knowledge-based models. Among the supervised systems, a linear classifier trained on conventional local features still proves to be a hard baseline to beat. Nonetheless, recent approaches exploiting neural networks on unlabeled corpora achieve promising results, surpassing this hard baseline in most test sets.

1 Introduction

Word Sense Disambiguation (WSD) has been a long-standing task in Natural Language Processing (NLP). It lies at the core of language understanding and has already been studied from many different angles (Navigli, 2009; Navigli, 2012). However, the field seems to be slowing down due to the lack of groundbreaking improvements and the difficulty of integrating current WSD systems into downstream NLP applications (de Lacalle and Agirre, 2015). In general the field does not have a clear path, partially owing to the fact that identifying real improvements over existing approaches becomes a hard task with current evaluation benchmarks. This is mainly due to the lack of a unified framework, which prevents direct and fair comparison among systems. Even though many evaluation datasets have been constructed for the task (Edmonds and Cotton, 2001; Snyder and Palmer, 2004; Navigli et al., 2007; Pradhan et al., 2007; Agirre et al., 2010a; Navigli et al., 2013; Moro and Navigli, 2015, *inter alia*), they tend to differ in format, construction guidelines and underlying sense inventory. In the case of the datasets annotated using WordNet (Miller, 1995), the *de facto* sense inventory for WSD, we encounter the additional barrier of having text annotated with different versions. These divergences are in the main solved individually by using or constructing automatic mappings. The quality check of such mapping, however, tends to be impractical and this leads to mapping errors which give rise to additional system inconsistencies in the experimental setting. This issue is directly extensible to the training corpora used by supervised systems. In fact, results obtained by supervised or semi-supervised systems reported in the literature are not completely reliable, because the systems may not necessarily have been trained on the same corpus, or the corpus was preprocessed differently, or annotated with a sense inventory different from the test data. Thus, together, the foregoing issues prevent us from drawing reliable conclusions on different models, as in some cases ostensible improvements may have been obtained as a consequence of the nature of the training corpus, the preprocessing pipeline or the version of the underlying sense inventory, rather than of the model itself. Moreover, because of these divergences, current systems tend to report results on a few datasets only, making it hard to perform a direct quantitative confrontation.

This paper offers two main contributions. First, we provide a complete evaluation framework for all-words Word Sense Disambiguation overcoming all the aforementioned limitations by (1) standardizing the WSD datasets and training corpora

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 99–110,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

into a unified format, (2) semi-automatically converting annotations from any dataset to WordNet 3.0, and (3) preprocessing the datasets by consistently using the same pipeline. Second, we use this evaluation framework to perform a fair quantitative and qualitative empirical comparison of the main techniques proposed in the WSD literature, including the latest advances based on neural networks.

2 State of the Art

The task of Word Sense Disambiguation consists of associating words in context with the most suitable entry in a pre-defined sense inventory. Depending on their nature, WSD systems are divided into two main groups: supervised and knowledge-based. In what follows we summarize the current state of these two types of approach.

2.1 Supervised WSD

Supervised models train different features extracted from manually sense-annotated corpora. These features have been mostly based on the information provided by the surroundings words of the target word (Keok and Ng, 2002; Navigli, 2009) and its collocations. Recently, more complex features based on word embeddings trained on unlabeled corpora have also been explored (Taghipour and Ng, 2015b; Rothe and Schütze, 2015; Iacobacci et al., 2016). These features are generally taken as input to train a linear classifier (Zhong and Ng, 2010; Shen et al., 2013). In addition to these conventional approaches, the latest developments in neural language models have motivated some researchers to include them in their WSD architectures (Kågebäck and Salomonsson, 2016; Melamud et al., 2016; Yuan et al., 2016).

Supervised models have traditionally been able to outperform knowledge-based systems (Navigli, 2009). However, obtaining sense-annotated corpora is highly expensive, and in many cases such corpora are not available for specific domains. This is the reason why some of these supervised methods have started to rely on unlabeled corpora as well. These approaches, which are often classified as *semi-supervised*, are targeted at overcoming the knowledge acquisition bottleneck of conventional supervised models (Pilehvar and Navigli, 2014). In fact, there is a line of research specifically aimed at automatically obtaining large amounts of high-quality sense-annotated corpora (Taghipour and Ng, 2015a; Raganato et al., 2016; Camacho-Collados et al., 2016a).

In this work we compare supervised systems and study the role of their underlying sense-annotated training corpus. Since semi-supervised models have been shown to outperform fully supervised systems in some settings (Taghipour and Ng, 2015b; Başkaya and Jurgens, 2016; Iacobacci et al., 2016; Yuan et al., 2016), we evaluate and compare models using both manually-curated and automatically-constructed sense-annotated corpora for training.

2.2 Knowledge-based WSD

In contrast to supervised systems, knowledge-based WSD techniques do not require any sense-annotated corpus. Instead, these approaches rely on the structure or content of manually-curated knowledge resources for disambiguation. One of the first approaches of this kind was Lesk (1986), which in its original version consisted of calculating the overlap between the context of the target word and its definitions as given by the sense inventory. Based on the same principle, various works have adapted the original algorithm by also taking into account definitions from related words (Banerjee and Pedersen, 2003), or by calculating the distributional similarity between definitions and the context of the target word (Basile et al., 2014; Chen et al., 2014). Distributional similarity has also been exploited in different settings in various works (Miller et al., 2012; Camacho-Collados et al., 2015; Camacho-Collados et al., 2016b). In addition to these approaches based on distributional similarity, an important branch of knowledge-based systems found their techniques on the structural properties of semantic graphs from lexical resources (Agirre and Soroa, 2009; Guo and Diab, 2010; Ponzetto and Navigli, 2010; Agirre et al., 2014; Moro et al., 2014; Weissenborn et al., 2015; Tripodi and Pelillo, 2016). Generally, these graph-based WSD systems first create a graph representation of the input text and then exploit different graph-based algorithms over the given representation (e.g., PageRank) to perform WSD.

3 Standardization of WSD datasets

In this section we explain our pipeline for transforming any given evaluation dataset or sense-annotated corpus into a preprocessed unified for-

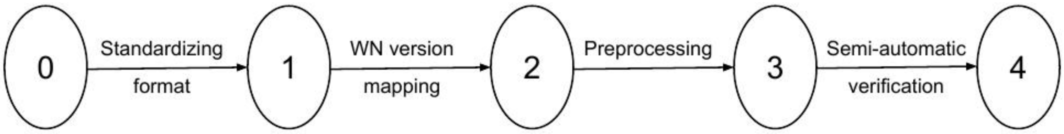

Figure 1: Pipeline for standardizing any given WSD dataset.

mat. In our pipeline we do not make any distinction between evaluation datasets and sense-annotated training corpora, as the pipeline can be applied equally to both types. For simplicity we will refer to both evaluation datasets and training corpora as WSD datasets.

Figure 1 summarizes our pipeline to standardize a WSD dataset. The process consists of four steps:

1. Most WSD datasets in the literature use a similar XML format, but they have some divergences on how to encode the information. For instance, the SemEval-15 dataset (Moro and Navigli, 2015) was developed for both WSD and Entity Linking and its format was especially designed for this latter task. Therefore, we decided to convert all datasets to a unified format. As unified format we use the XML scheme used for the SemEval-13 all-words WSD task (Navigli et al., 2013), where preprocessing information of a given corpus is also encoded.

2. Once the dataset is converted to a unified format, we map the sense annotations from its original WordNet version to 3.0, which is the latest version of WordNet used in evaluation datasets. This mapping is carried out semi-automatically. First, we use automatically-constructed WordNet mappings[1] (Daude et al., 2003). These mappings provide confidence values which we use to initially map senses whose mapping confidence is 100%. Then, the annotations of the remaining senses are manually checked, and re-annotated or removed whenever necessary[2]. Additionally, in this step we decided to remove all annotations of auxiliary verbs, following the annotation guidelines of the latest WSD datasets.

3. The third step consists of preprocessing the given dataset. We used the Stanford CoreNLP toolkit (Manning et al., 2014) for Part-of-Speech (PoS) tagging[3] and lemmatization. This step is performed in order to ensure that all systems use the same preprocessed data.

4. Finally, we developed a script to check that the final dataset conforms to the aforementioned guidelines. In this final verification we also ensured that the sense annotations match the lemma and the PoS tag provided by Stanford CoreNLP by automatically fixing all divergences.

4 Data

In this section we summarize the WSD datasets used in the evaluation framework. To all these datasets we apply the standardization pipeline described in Section 3. First, we enumerate all the datasets used for the evaluation (Section 4.1). Second, we describe the sense-annotated corpora used for training (Section 4.2). Finally, we show some relevant statistics extracted from these resources (Section 4.3).

4.1 WSD evaluation datasets

For our evaluation framework we considered five standard all-words fine-grained WSD datasets from the Senseval and SemEval competitions:

- **Senseval-2** (Edmonds and Cotton, 2001). This dataset was originally annotated with WordNet 1.7. After standardization, it consists of 2282 sense annotations, including nouns, verbs, adverbs and adjectives.

- **Senseval-3 task 1** (Snyder and Palmer, 2004). The WordNet version of this dataset was 1.7.1. It consists of three documents from three different domains (editorial, news story and fiction), totaling 1850 sense annotations.

[1] `http://nlp.lsi.upc.edu/tools/download-map.php`

[2] This manual correction involved less than 10% of all instances for the datasets for which this step was performed.

[3] In order to have a standard format which may be used by languages other than English, we provide coarse-grained PoS tags as given by the universal PoS tagset (Petrov et al., 2011).

	#Docs	#Sents	#Tokens	#Annotations	#Sense types	#Word types	Ambiguity
Senseval-2	3	242	5,766	2,282	1,335	1,093	5.4
Senseval-3	3	352	5,541	1,850	1,167	977	6.8
SemEval-07	3	135	3,201	455	375	330	8.5
SemEval-13	13	306	8,391	1,644	827	751	4.9
SemEval-15	4	138	2,604	1,022	659	512	5.5
SemCor	352	37,176	802,443	226,036	33,362	22,436	6.8
OMSTI	-	813,798	30,441,386	911,134	3,730	1,149	8.9

Table 1: Statistics of the WSD datasets used in the evaluation framework (after standardization).

- **SemEval-07 task 17** (Pradhan et al., 2007). This is the smallest among the five datasets, containing 455 sense annotations for nouns and verbs only. It was originally annotated using WordNet 2.1 sense inventory.

- **SemEval-13 task 12** (Navigli et al., 2013). This dataset includes thirteen documents from various domains. In this case the original sense inventory was WordNet 3.0, which is the same as the one that we use for all datasets. The number of sense annotations is 1644, although only nouns are considered.

- **SemEval-15 task 13** (Moro and Navigli, 2015). This is the most recent WSD dataset available to date, annotated with WordNet 3.0. It consists of 1022 sense annotations in four documents coming from three heterogeneous domains: biomedical, mathematics/computing and social issues.

4.2 Sense-annotated training corpora

We now describe the two WordNet sense-annotated corpora used for training the supervised systems in our evaluation framework:

- **SemCor** (Miller et al., 1994). SemCor[4] is a manually sense-annotated corpus divided into 352 documents for a total of 226,040 sense annotations. It was originally tagged with senses from the WordNet 1.4 sense inventory. SemCor is, to our knowledge, the largest corpus manually annotated with WordNet senses, and is the main corpus used in the literature to train supervised WSD systems (Agirre et al., 2010b; Zhong and Ng, 2010).

- **OMSTI** (Taghipour and Ng, 2015a). OMSTI (*One Million Sense-Tagged Instances*) is a large corpus annotated with senses from the WordNet 3.0 inventory. It was automatically constructed by using an alignment-based WSD approach (Chan and Ng, 2005) on a large English-Chinese parallel corpus (Eisele and Chen, 2010, MultiUN corpus). OMSTI[5] has already shown its potential as a training corpus by improving the performance of supervised systems which add it to existing training data (Taghipour and Ng, 2015a; Iacobacci et al., 2016).

4.3 Statistics

Table 1 shows some statistics[6] of the WSD datasets and training corpora which we use in the evaluation framework. The number of sense annotations varies across datasets, ranging from 455 annotations in the SemEval-07 dataset, to 2,282 annotations in the Senseval-2 dataset. As regards sense-annotated corpora, OMSTI is made up of almost 1M sense annotations, a considerable increase over the number of sense annotations of SemCor. However, SemCor is much more balanced in terms of unique senses covered (3,730 covered by OMSTI in contrast to over 33K covered by SemCor). Additionally, while OMSTI was constructed automatically, SemCor was manually built and, hence, its quality is expected to be higher.

Finally, we calculated the ambiguity level of each dataset, computed as the total number of can-

[4] We downloaded the SemCor 3.0 version at `web.eecs.umich.edu/~mihalcea/downloads.html`

[5] In this paper we refer to the portion of sense-annotated data from the MultiUN corpus as OMSTI. Note that OMSTI was released along with SemCor.

[6] Statistics included in Table 1: number of documents (#Docs), sentences (#Sents), tokens (#Tokens), sense annotations (#Annotations), sense types covered (#Sense types), annotated lemma types covered (#Word types), and ambiguity level (Ambiguity). There was no document information in the OMSTI data released by Taghipour and Ng (2015a).

didate senses (i.e., senses sharing the surface form of the target word) divided by the number of sense annotations. The highest ambiguity is found on OMSTI, which, despite being constructed automatically, contains a high coverage of ambiguous words. As far as the evaluation competition datasets are concerned, the ambiguity may give a hint as to how difficult a given dataset may be. In this case, SemEval-07 displays the highest ambiguity level among all evaluation datasets.

5 Evaluation

The evaluation framework consists of the WSD evaluation datasets described in Section 4.1. In this section we use this framework to perform an empirical comparison among a set of heterogeneous WSD systems. The systems used in the evaluation are described in detail in Section 5.1, the results are shown in Section 5.2 and a detailed analysis is presented in Section 5.3.

5.1 Comparison systems

We include three supervised (Section 5.1.1) and three knowledge-based (Section 5.1.2) all-words WSD systems in our empirical comparison.

5.1.1 Supervised

To ensure a fair comparison, all supervised systems use the same corpus for training: SemCor and Semcor+OMSTI[7] (see Section 4.2). In the following we describe the three supervised WSD systems used in the evaluation:

- **IMS** (Zhong and Ng, 2010) uses a Support Vector Machine (SVM) classifier over a set of conventional WSD features. IMS[8] is built on a flexible framework which allows an easy integration of different features. The default implementation includes surrounding words, PoS tags of surroundings words, and local collocations as features.

- **IMS+embeddings** (Taghipour and Ng, 2015b; Rothe and Schütze, 2015; Iacobacci et al., 2016). These approaches have shown the potential of using word embeddings on the WSD task. Iacobacci et al. (2016) carried out a comparison of different strategies for integrating word embeddings as a feature in WSD. In this paper we consider the two best configurations in Iacobacci et al. (2016)[9]: using all IMS default features including and excluding surrounding words (IMS+emb and IMS$_{\text{-s}}$+emb, respectively). In both cases word embeddings are integrated using exponential decay (i.e., word weights drop exponentially as the distance towards the target word increases). Likewise, we use Iacobacci et al.'s suggested learning strategy and hyperparameters to train the word embeddings: Skip-gram model of Word2Vec[10] (Mikolov et al., 2013) with 400 dimensions, ten negative samples and a window size of ten words. As unlabeled corpus to train the word embeddings we use the English ukWaC corpus[11] (Baroni et al., 2009), which is made up of two billion words from paragraphs extracted from the web.

- **Context2Vec** (Melamud et al., 2016). Neural language models have recently shown their potential for the WSD task (Kågebäck and Salomonsson, 2016; Yuan et al., 2016). In this experiment we replicated the approach of Melamud et al. (2016, Context2Vec), for which the code[12] is publicly available. This approach is divided in three steps. First, a bidirectional LSTM recurrent neural network is trained on an unlabeled corpus (we considered the same ukWaC corpus used by the previous comparison system). Then, a context vector is learned for each sense annotation in the training corpus. Finally, the sense annotation whose context vector is closer to the target word's context vector is selected as the intended sense.

Finally, as baseline we included the Most Frequent Sense (**MFS**) heuristic, which for each target word selects the sense occurring the highest number of times in the training corpus.

[7] As already noted by Taghipour and Ng (2015a), supervised systems trained on only OMSTI obtain lower results than when trained along with SemCor, mainly due to OMSTI's lack of coverage in target word types.

[8] We used the original implementation available at `http://www.comp.nus.edu.sg/~nlp/software.html`

[9] We used the implementation available at `https://github.com/iiacobac/ims_wsd_emb`

[10] `code.google.com/archive/p/word2vec/`

[11] `http://wacky.sslmit.unibo.it/doku.php?id=corpora`

[12] `https://github.com/orenmel/context2vec`

5.1.2 Knowledge-based

In this section we describe the three knowledge-based WSD models used in our empirical comparison:

- **Lesk** (Lesk, 1986) is a simple knowledge-based WSD algorithm that bases its calculations on the overlap between the definitions of a given sense and the context of the target word. For our experiments we replicated the extended version of the original algorithm in which definitions of related senses are also considered and the conventional term frequency-inverse document frequency (Jones, 1972, *tf-idf*) is used for word weighting (Banerjee and Pedersen, 2003, Lesk$_{\text{ext}}$). Additionally, we included the enhanced version of Lesk in which word embeddings[13] are leveraged to compute the similarity between definitions and the target context (Basile et al., 2014, Lesk$_{\text{ext}}$+emb)[14].

- **UKB** (Agirre and Soroa, 2009; Agirre et al., 2014) is a graph-based WSD system which makes use of random walks over a semantic network (WordNet graph in this case). UKB[15] applies the Personalized Page Rank algorithm (Haveliwala, 2002) initialized using the context of the target word. Unlike most WSD systems, UKB does not back-off to the WordNet first sense heuristic and it is self-contained (i.e., it does not make use of any external resources/corpora). We used both default configurations from UKB: using the full WordNet graph (UKB) and the full graph including disambiguated glosses as connections as well (UKB_gloss).

- **Babelfy** (Moro et al., 2014) is a graph-based disambiguation approach which exploits random walks to determine connections between synsets. Specifically, Babelfy[16] uses random walks with restart (Tong et al., 2006) over BabelNet (Navigli and Ponzetto, 2012), a large semantic network integrating WordNet among other resources such as Wikipedia or Wiktionary. Its algorithm is based on a densest subgraph heuristic for selecting high-coherence semantic interpretations of the input text. The best configuration of Babelfy takes into account not only the target sentence in which the target word occurs, but also the whole document.

As knowledge-based baseline we included the **WordNet first sense**. This baseline simply selects the candidate which is considered as first sense in WordNet 3.0. Even though the sense order was decided on the basis of semantically-tagged text, we considered it as knowledge-based in this experiment as this information is already available in WordNet. In fact, knowledge-based systems like Babelfy include this information in their pipeline. Despite its simplicity, this baseline has been shown to be hard to beat by automatic WSD systems (Navigli, 2009; Agirre et al., 2014).

5.2 Results

Table 2 shows the F-Measure performance of all comparison systems on the five all-words WSD datasets. Since not all test word instances are covered by the corresponding training corpora, supervised systems have a maximum F-Score (*ceiling* in the Table) they can achieve. Nevertheless, supervised systems consistently outperform knowledge-based systems across datasets, confirming the results of Pilehvar and Navigli (2014). A simple linear classifier over conventional WSD features (i.e., IMS) proves to be robust across datasets, consistently outperforming the MFS baseline. The recent integration of word embeddings as an additional feature is beneficial, especially as a replacement of the feature based on the surface form of surrounding words (i.e., IMS$_{\text{-s}}$+emb). Moreover, recent advances on neural language models (in the case of Context2Vec a bi-directional LSTM) appear to be highly promising for the WSD task according to the results, as Context2Vec outperforms IMS in most datasets.

On the other hand, it is also interesting to note the performance inconsistencies of systems across datasets, as in all cases there is a large performance gap between the best and the worst performing dataset. As explained in Section 4.3, the ambiguity level may give a hint as to how difficult the corresponding dataset may be. In fact, WSD systems obtain relatively low results in SemEval-07, which is the most ambiguous dataset (see Table 1).

[13] We used the same word embeddings described in Section 5.1.1 for IMS+emb.

[14] We used the implementation from `https://github.com/pippokill/lesk-wsd-dsm`. In this implementation additional definitions from BabelNet are considered.

[15] We used the last implementation available at `http://ixa2.si.ehu.es/ukb/`

[16] We used the Java API from `http://babelfy.org`

	Tr. Corpus	System	Senseval-2	Senseval-3	SemEval-07	SemEval-13	SemEval-15
Supervised	**SemCor**	IMS	70.9	69.3	61.3	65.3	69.5
		IMS+emb	71.0	69.3	60.9	**67.3**	71.3
		IMS$_{\text{-s}}$+emb	**72.2**	**70.4**	**62.6**	65.9	71.5
		Context2Vec	71.8	69.1	61.3	65.6	**71.9**
		MFS	65.6	66.0	54.5	63.8	67.1
		Ceiling	*91.0*	*94.5*	*93.8*	*88.6*	*90.4*
	SemCor + OMSTI	IMS	72.8	69.2	60.0	65.0	69.3
		IMS+emb	70.8	68.9	58.5	66.3	69.7
		IMS$_{\text{-s}}$+emb	**73.3**	**69.6**	61.1	66.7	70.4
		Context2Vec	72.3	68.2	**61.5**	**67.2**	**71.7**
		MFS	66.5	60.4	52.3	62.6	64.2
		Ceiling	*91.5*	*94.9*	*94.7*	*89.6*	*91.1*
Knowledge	**-**	Lesk$_{\text{ext}}$	50.6	44.5	32.0	53.6	51.0
		Lesk$_{\text{ext}}$+emb	63.0	63.7	**56.7**	66.2	64.6
		UKB	56.0	51.7	39.0	53.6	55.2
		UKB_gloss	60.6	54.1	42.0	59.0	61.2
		Babelfy	**67.0**	63.5	51.6	**66.4**	**70.3**
		WN 1$^{\text{st}}$ sense	66.8	**66.2**	55.2	63.0	67.8

Table 2: F-Measure percentage of different models in five all-words WSD datasets.

	Nouns	Verbs	Adj.	Adv.	All
#Instances	4,300	1,652	955	346	7,253
Ambiguity	4.8	10.4	3.8	3.1	5.8

Table 3: Number of instances and ambiguity level of the concatenation of all five WSD datasets.

However, this is the dataset in which supervised systems achieve a larger margin with respect to the MFS baseline, which suggests that, in general, the MFS heuristic does not perform accurately on highly ambiguous words.

5.3 Analysis

To complement the results from the previous section, we additionally carried out a detailed analysis about the global performance of each system and divided by PoS tag. To this end, we concatenated all five datasets into a single dataset. This resulted in a large evaluation dataset of 7,253 instances to disambiguate (see Table 3). Table 4 shows the F-Measure performance of all comparison systems on the concatenation of all five WSD evaluation datasets, divided by PoS tag. IMS$_{\text{-s}}$+emb trained on SemCor+OMSTI achieves the best overall results, slightly above Context2Vec trained on the same corpus. In what follows we describe some of the main findings extracted from our analysis.

Training corpus. In general, the results of supervised systems trained on SemCor only (manually-annotated) are lower than training simultaneously on both SemCor and OMSTI (automatically-annotated). This is a promising finding, which confirms the results of previous works (Raganato et al., 2016; Iacobacci et al., 2016; Yuan et al., 2016) and encourages further research on developing reliable automatic or semi-automatic methods to obtain large amounts of sense-annotated corpora in order to overcome the knowledge-acquisition bottleneck. For instance, Context2Vec improves 0.4 points overall when adding the automatically sense-annotated OMSTI as part of the training corpus, suggesting that more data, even if not perfectly clean, may be beneficial for neural language models.

Knowledge-based vs. Supervised. One of the main conclusions that can be taken from the evaluation is that supervised systems clearly outperform knowledge-based models. This may be due to the fact that in many cases the main disambiguation clue is given by the immediate local context. This is particularly problematic for knowledge-based systems, as they take equally into account all the words within a sentence (or document in the case of Babelfy). For instance, in the following sentence, both UKB and Babelfy fail to predict the correct sense of *state*:

In sum, at both the federal and ***state*** *government levels at least part of the seemingly irrational behavior voters display in the voting booth may have an exceedingly rational explanation.*

	Tr. Corpus	System	Nouns	Verbs	Adjectives	Adverbs	All
Supervised	**SemCor**	IMS	70.4	56.1	75.6	82.9	68.4
		IMS+emb	71.8	55.4	**76.1**	82.7	69.1
		IMS$_{-s}$+emb	**71.9**	56.9	75.9	**84.7**	**69.6**
		Context2Vec	71.0	**57.6**	75.2	82.7	69.0
		MFS	67.6	49.6	73.1	80.5	64.8
		Ceiling	*89.6*	*95.1*	*91.5*	*96.4*	*91.5*
	SemCor + OMSTI	IMS	70.5	**56.9**	76.8	82.9	68.8
		IMS+emb	71.0	53.3	77.1	82.7	68.3
		IMS$_{-s}$+emb	**72.0**	56.5	76.6	**84.7**	**69.7**
		Context2Vec	71.7	55.8	**77.2**	82.7	69.4
		MFS	65.8	45.9	72.7	80.5	62.9
		Ceiling	*90.4*	*95.8*	*91.8*	*96.4*	*92.1*
Knowledge	**-**	Lesk$_{ext}$	54.1	27.9	54.6	60.3	48.7
		Lesk$_{ext}$+emb	**69.8**	**51.2**	51.7	80.6	63.7
		UKB	56.7	39.3	63.9	44.0	53.2
		UKB_gloss	62.1	38.3	66.8	66.2	57.5
		Babelfy	68.6	49.9	73.2	79.8	**65.5**
		WN 1st sense	67.6	50.3	**74.3**	**80.9**	65.2

Table 4: F-Measure percentage of different models on the concatenation of all five WSD datasets.

In this sentence, *state* is annotated with its *administrative districts of a nation* sense in the gold standard. The main disambiguation clue seems to be given by its previous and immediate subsequent words (*federal* and *government*), which tend to co-occur with this particular sense. However, knowledge-based WSD systems like UKB or Babelfy give the same weight to all words in context, underrating the importance of this local disambiguation clue in the example. For instance, UKB disambiguates *state* with the sense defined as *the way something is with respect to its main attributes*, probably biased by words which are not immediately next to the target word within the sentence, e.g., *irrational*, *behaviour*, *rational* or *explanation*.

Low overall performance on verbs. As can be seen from Table 4, the F-Measure performance of all systems on verbs is in all cases below 58%. This can be explained by the high granularity of verbs in WordNet. For instance, the verb *keep* consists of 22 different meanings in WordNet 3.0, six of them denoting "possession and transfer of possession"[17]. In fact, the average ambiguity level of all verbs in this evaluation framework is 10.4 (see Table 3), considerably greater than the ambiguity on other PoS tags, e.g., 4.8 in nouns. Nonetheless, supervised systems manage to comfortably outperform the MFS baseline, which does not seem to be reliable for verbs given their high ambiguity.

Influence of preprocessing. As mentioned in Section 3, our evaluation framework provides a preprocessing of the corpora with Stanford CoreNLP. This ensures a fair comparison among all systems but may introduce some annotation inaccuracies, such as erroneous PoS tags. However, for English these errors are minimal[18]. For instance, the global error rate of the Stanford PoS tagger in all disambiguation instances is 3.9%, which were fixed as explained in Section 3.

Bias towards the Most Frequent Sense. After carrying out an analysis on the influence of MFS in WSD systems[19], we found that all supervised systems suffer a strong bias towards the MFS, with all IMS-based systems disambiguating over 75% of instances with their MFS. Context2Vec is slightly less affected by this bias, with 71.5% (SemCor) and 74.7% (SemCor+OMSTI) of answers corre-

[17] `https://wordnet.princeton.edu/man/lexnames.5WN.html`

[18] Even if preprocessing plays a minimal role for English, it may be of higher importance for other languages, e.g., morphologically richer languages (Eger et al., 2016).

[19] See Postma et al. (2016) for an interesting discussion on the bias of current WSD systems towards the MFS.

sponding to the MFS. Interestingly, this MFS bias is also present in graph knowledge-based systems. In fact, Calvo and Gelbukh (2015) had already shown how the MFS correlates strongly with the number of connections in WordNet.

Knowledge-based systems. For knowledge-based systems the WN first sense baseline proves still to be extremely hard to beat. The only knowledge-based system that overall manages to beat this baseline is Babelfy, which, in fact, uses information about the first sense in its pipeline. Babelfy's default pipeline includes a confidence threshold in order to decide whether to disambiguate or back-off to the first sense. In total, Babelfy backs-off to WN first sense in 63% of all instances. Nonetheless, it is interesting to note the high performance of Babelfy and Lesk_{ext}+emb on noun instances (outperforming the first sense baseline by 1.0 and 2.2 points, respectively) in contrast to their relatively lower performance on verbs, adjectives[20] and adverbs. We believe that this is due to the nature of the lexical resource used by these two systems, i.e., BabelNet. BabelNet includes Wikipedia as one of its main sources of information. However, while Wikipedia provides a large amount of semantic connections and definitions for nouns, this it not the case for verbs, adjectives and adverbs, as they are not included in Wikipedia and their source of information mostly comes from WordNet only.

6 Conclusion and Future Work

In this paper we presented a unified evaluation framework for all-words WSD. This framework is based on evaluation datasets taken from Senseval and SemEval competitions, as well as manually and automatically sense-annotated corpora. In this evaluation framework all datasets share a common format, sense inventory (i.e., WordNet 3.0) and preprocessing pipeline, which eases the task of researchers to evaluate their models and, more importantly, ensures a fair comparison among all systems. The whole evaluation framework[21], including guidelines for researchers to include their own sense-annotated datasets and a script to validate their conformity to the guidelines, is available at `http://lcl.uniroma1.it/wsdeval`.

We used this framework to perform an empirical comparison among a set of heterogeneous WSD systems, including both knowledge-based and supervised ones. Supervised systems based on neural networks achieve the most promising results. Given our analysis, we foresee two potential research avenues focused on semi-supervised learning: (1) exploiting large amounts of unlabeled corpora for learning word embeddings or training neural language models, and (2) automatically constructing high-quality sense-annotated corpora to be used by supervised WSD systems. As far as knowledge-based systems are concerned, enriching knowledge resources with semantic connections for non-nominal mentions may be an important step towards improving their performance.

For future work we plan to further extend our unified framework to languages other than English, including SemEval multilingual WSD datasets, as well as to other sense inventories such as Open Multilingual WordNet, BabelNet and Wikipedia, which are available in different languages.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.

Jose Camacho-Collados is supported by a Google PhD Fellowship in Natural Language Processing.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of EACL*, pages 33–41.

Eneko Agirre, Oier Lopez De Lacalle, Christiane Fellbaum, Andrea Marchetti, Antonio Toral, and Piek Vossen. 2010a. Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 123–128.

Eneko Agirre, Oier Lopez De Lacalle, Christiane Fellbaum, Andrea Marchetti, Antonio Toral, and Piek Vossen. 2010b. Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 123–128.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word

[20] The poor performance of Lesk_{ext}+emb on adjective instances is particularly noticeable.

[21] We have additionally set up a CodaLab competition based on this evaluation framework.

sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlap as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, Mexico.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland.

Osman Başkaya and David Jurgens. 2016. Semi-supervised learning with induced word senses for state of the art word sense disambiguation. *Journal of Artificial Intelligence Research*, 55:1025–1058.

Hiram Calvo and Alexander Gelbukh. 2015. Is the most frequent sense of a word better connected in a semantic network? In *International Conference on Intelligent Computing*, pages 491–499. Springer.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. A Unified Multilingual Semantic Representation of Concepts. In *Proceedings of ACL*, pages 741–751.

José Camacho-Collados, Claudio Delli Bovi, Alessandro Raganato, and Roberto Navigli. 2016a. A Large-Scale Multilingual Disambiguation of Glosses. In *Proceedings of LREC*, pages 1701–1708, Portoroz, Slovenia.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016b. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.

Yee Seng Chan and Hwee Tou Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *AAAI*, volume 5, pages 1037–1042.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of EMNLP*, pages 1025–1035, Doha, Qatar.

Jordi Daude, Lluis Padro, and German Rigau. 2003. Validation and tuning of wordnet mapping techniques. In *Proceedings of RANLP*.

Oier Lopez de Lacalle and Eneko Agirre. 2015. A methodology for word sense disambiguation at 90% based on large-scale crowdsourcing. *Lexical and Computational Semantics (* SEM 2015)*, page 61.

Philip Edmonds and Scott Cotton. 2001. Senseval-2: Overview. In *Proceedings of The Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–6, Toulouse, France.

Steffen Eger, Rüdiger Gleim, and Alexander Mehler. 2016. Lemmatization and morphological tagging in german and latin: A comparison and a survey of the state-of-the-art. In *Proceedings of LREC 2016*.

Andreas Eisele and Yu Chen. 2010. MultiUN: A Multilingual Corpus from United Nation Documents. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 2868–2872.

Weiwei Guo and Mona T. Diab. 2010. Combining orthogonal monolingual and multilingual sources of evidence for all words WSD. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1542–1551, Uppsala, Sweden.

Taher H. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, pages 517–526, Hawaii, USA.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of ACL*, pages 897–907, Berlin, Germany.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional lstm. *arXiv preprint arXiv:1606.03568*.

L. Y. Keok and H. T. Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the 7^{th} Conference on Empirical Methods in Natural Language Processing*, pages 41–48, Philadelphia, USA.

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 Conference on Systems Documentation,* Toronto, Ontario, Canada, pages 24–26.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of CONLL*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the workshop on Human Language Technology*, pages 240–243. Association for Computational Linguistics.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *COLING*, pages 1781–1796.

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

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

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

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. SemEval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic*, pages 30–35.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of SemEval 2013*, pages 222–231.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Roberto Navigli. 2012. A quick tour of word sense disambiguation, induction and related approaches. In *SOFSEM 2012: Theory and practice of computer science*, pages 115–129. Springer.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Mohammad Taher Pilehvar and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art Word Sense Disambiguation. *Computational Linguistics*, 40(4).

Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich Word Sense Disambiguation rivaling supervised system. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1522–1531, Uppsala, Sweden.

Marten Postma, Ruben Izquierdo, Eneko Agirre, German Rigau, and Piek Vossen. 2016. Addressing the MFS Bias in WSD systems. In *Proceedings of LREC*, Portoroz, Slovenia.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval*, pages 87–92.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2016. Automatic Construction and Evaluation of a Large Semantically Enriched Wikipedia. In *Proceedings of IJCAI*, pages 2894–2900, New York City, NY, USA, July.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of ACL*, pages 1793–1803, Beijing, China.

Hui Shen, Razvan Bunescu, and Rada Mihalcea. 2013. Coarse to fine grained sense disambiguation in wikipedia. *Proc. of *SEM*, pages 22–31.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3), Barcelona, Spain*, pages 41–43, Barcelona, Spain.

Kaveh Taghipour and Hwee Tou Ng. 2015a. One million sense-tagged instances for word sense disambiguation and induction. *CoNLL 2015*, page 338.

Kaveh Taghipour and Hwee Tou Ng. 2015b. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. *Proceedings of NAACL HLT 2015*, pages 314–323.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *ICDM*, pages 613–622.

Rocco Tripodi and Marcello Pelillo. 2016. A game-theoretic approach to word sense disambiguation. *arXiv preprint arXiv:1606.07711*.

Dirk Weissenborn, Leonhard Hennig, Feiyu Xu, and Hans Uszkoreit. 2015. Multi-Objective Optimization for the Joint Disambiguation of Nouns and Named Entities. In *Proceedings of ACL*, pages 596–605, Beijing, China.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING*, pages 1374–1385.

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 System Demonstrations*, pages 78–83.

Similar but not the Same: Word Sense Disambiguation Improves Event Detection via Neural Representation Matching

Weiyi Lu[†] **and Thien Huu Nguyen**[#‡]
[†] Computer Science Department, New York University, USA
[#] Montreal Institute for Learning Algorithms, University of Montreal, Canada
[‡] Department of Computer and Information Science, University of Oregon, USA
weiyi.lu@nyu.edu,thien@cs.uoregon.edu

Abstract

Event detection (ED) and word sense disambiguation (WSD) are two similar tasks in that they both involve identifying the classes (i.e. event types or word senses) of some word in a given sentence. It is thus possible to extract the knowledge hidden in the data for WSD, and utilize it to improve the performance on ED. In this work, we propose a method to transfer the knowledge learned on WSD to ED by matching the neural representations learned for the two tasks. Our experiments on two widely used datasets for ED demonstrate the effectiveness of the proposed method.

1 Introduction

An important aspect of natural language processing involves understanding events mentioned in text. Towards this end, event detection (ED) is the task of locating event triggers (usually verbs or nouns) within a given text, and classifying them among a given set of event types. This task remains challenging due to the inherent ambiguity and flexibility of natural languages. The current state-of-the-art methods for ED have involved applying deep learning (DL) models to automatically extract feature representations of the text, and then treating the task as a classification problem (Chen et al., 2015; Nguyen and Grishman, 2015b).

The major intuition in this paper is that the task of ED is closely related to the task of word sense disambiguation (WSD) whose datasets can help to improve the performance of the DL models for ED. This is due to the goal of WSD to determine the sense of a word within a particular context, given a set of possible senses that the word can take on. Our intuition is based on the two following aspects:

(i) Similar Context Modeling: Given a word in a context/sentence, both ED and WSD models need to select/predict a correct label in a list of candidate labels for the word. For WSD, the candidate labels are the possible senses (e.g, sense ids in WordNet) that the word of interest can have, while for ED, they are the set of predetermined event types (e.g, the event subtypes in the ACE 2005 dataset[1]). Consider the word "*fired*" in the following sentence as an example:

The boss ***fired*** *his secretary today.*

For WSD, there are 12 possible senses for the verb "*fire*" in WordNet in which the correct label for the word "*fired*" in this case is the sense id "*fire%2:41:00::*" (i.e, "*terminate the employment of*"). The ED task in the ACE 2005 dataset, on the other hand, involves 33 possible event subtypes with "*End-Position*" as the correct event subtype/label for the word "*fired*" in our example.

In order to make such label predictions, both ED and WSD need to model the word itself and its context (i.e, the words "*fired*", "*boss*", and "*secretary*" in the example). This similar modeling allows the same DL model to be adopted for both ED and WSD, facilitating the use of WSD data to improve the feature representations for ED via parameter/representation tying.

(ii) Close Semantic Consideration: As there are some overlaps between the semantic differentiation in WSD and ED, the knowledge/information from WSD about a particular word in a context can help to make a better prediction for that word in ED. For instance, in the example above, the knowledge from WSD that the word "*fired*" is referring to a termination of employment would clearly help ED to identify "*End-Position*" as the correct event type (rather than the incorrect event type "*Attack*") for "*fired*" in this case.

How can we exploit this intuition to improve the performance of the DL models for ED with WSD

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4822–4828
Brussels, Belgium, October 31 - November 4, 2018.

data? In this work, we propose a novel method based on representation matching to transfer the knowledge learned from the WSD data to the DL models for ED. In particular, two separate deep learning models are employed to model the context for WSD and ED. The two models share the network architecture, but involve different parameters that are specific to the tasks. We then transfer the knowledge from the WSD network to the ED network by ensuring that the feature representations learned by the two networks on the same contexts are similar to each other.

We demonstrate the effectiveness of the proposed method on two widely used datasets for ED. To the best of our knowledge, this is the first work to study the transfer learning/multi-task learning methods for WSD and ED with DL.

2 Model

We consider the typical setting where we have two separate datasets $D^{wsd} = \{W_i^{wsd}, p_i^{wsd}, y_i^{wsd}\}$ for WSD and $D^{ed} = \{W_i^{ed}, p_i^{ed}, y_i^{ed}\}$ for ED. Here, W_i^{ed} is the i-the sentence of D^{ed}, p_i^{ed} is the index of the word of interest for event type prediction in W_i^{ed}, and y_i^{ed} is the corresponding event type label. The same conventions apply for $W_i^{wsd}, p_i^{wsd}, y_i^{wsd}$. Also, let Y^{wsd} and Y^{ed} be the label sets for WSD and ED respectively (i.e, $y_i^{wsd} \in Y^{wsd}$ and $y_i^{ed} \in Y^{ed}$). Our goal is to transfer the knowledge learned from the D^{wsd} dataset to improve the performance of the ED models trained on the D^{ed} dataset (multi-task learning).

In the following, we will first describe the deep learning architectures to transform the sentences W in the datasets D^{wsd} and D^{ed} into representation vectors. We only focus on the deep learning architectures proposed for ED in the literature to achieve compatible comparisons for ED. The proposed multi-task learning method for ED with the WSD dataset will follow.

2.1 Computing the Feature Representations

Consider a sentence W in the datasets D^{wsd} or D^{ed} that is represented as a sequence of tokens $W = [w_0, w_1, \ldots, w_t]$. Let p be the index of the word of interest in this sentence. The context for w_p in W is constructed by taking the word itself, the n preceding words, and the n following words (padding or truncating when necessary). The tokens in the context are re-indexed to form an instance $V = [v_0, v_1, \ldots, v_n, \ldots, v_{2n-1}, v_{2n}]$, where v_n corresponds to w_p in W.

Encoding

The first step to prepare the instance V for the deep learning models is to map each token v_j in V into two real-valued vectors, which are then concatenated to form a vector representation x_j for v_j (Nguyen and Grishman, 2015b; Chen et al., 2015):

1. The word embedding of v_j obtained by looking up the token v_j in the pre-trained word embedding table (Mikolov et al., 2013a).

2. The position embedding vector for v_j: obtained by looking up the relative distance $j - n$ of v_j with respect to the token of interest v_n in a position embedding table (randomly initialized) (Chen et al., 2015; Nguyen and Grishman, 2015a).

It is important to note that, different from the prior works (Nguyen and Grishman, 2015b; Liu et al., 2017), we do not include the entity type label of each token into its representation. This is a more realistic setting for our work as the golden entity mentions do not always exist in practice, especially for the datasets in WSD.

Once each token v_j is converted into the representation vector x_j, the instance V becomes a sequence of vectors $X = [x_0, x_1, \ldots, x_n, \ldots, x_{2n-1}, x_{2n}]$ that would be fed into the one of the following deep learning models to learn a feature representation R for V.

Typical Deep Learning Models for ED

1. *CNN*: This is the convolutional neural networks in(Nguyen and Grishman, 2015b; Chen et al., 2015). It features convolution operations that are performed over the k consecutive vectors (k-grams) in X and followed by a max-pooling layer to generate the representation vector R for V. Multiple window values k are used to enhance the coverage of the model over the hidden k-grams in the context.

2. *NCNN* (Nguyen and Grishman, 2016d): This model is similar to *CNN*. The only difference is instead of running the convolution over the k consecutive vectors, *NCNN* convolutes over the k arbitrarily non-consecutive k vectors in V. This helps *NCNN* to explicitly model the non-consecutive words in the context to improve ED.

3. *BiRNN*: This is the bidirectional recurrent neural network (RNN) for event extraction in (Nguyen et al., 2016a). The model is

composed of two recurrent neural networks (RNN), where one runs forward and the other runs backward through the input sequence V. The hidden vectors produced by the two networks are then concatenated at each position in the context. The vector at the position of n for the word of interest is used as the representation vector R for V. Due to the property of RNN, R encodes the information over the whole input V with a greater focus on v_n.

4. *CNN+BiRNN*: In this model (Feng et al., 2016), X is passed through both a *CNN* and a *BiRNN* whose results are concatenated to produce the hidden representation R for ED. The expectation is to take advantage of the modeling abilities from both the *CNN* and *BiRNN* architectures for ED.

In practice, the representation vector R (obtained from one of the deep learning models above) is also concatenated with the word embeddings of the tokens surrounding the token of interest w_n to improve its expressiveness (Chen et al., 2015; Nguyen and Grishman, 2016d). We would use this extended version when we refer to R in the following.

In the final step, the representation vector R is fed into a feed-forward neural network followed by a softmax layer to perform predictions for ED and WSD.

For convenience, we denote the whole process that a DL model M is used to compute the representation vector R for the input sentence W with the token index p of interest as: $R = M(W, p)$.

2.2 Multi-task Learning Models

The previous section has described the deep learning methods that can be employed to train the models for ED and WSD separately. This section presents our proposed method to transfer the knowledge from the WSD dataset to improve the performance for ED.

A typical method for transfer learning/multi-task learning in NLP is to alternate the training process for the parameter-shared models of the related tasks (possibly with different datasets) (Guo et al., 2016; Li et al., 2015; Liu et al., 2016). For instance, in (Guo et al., 2016), the authors use the same deep learning model to learn the feature representations for the text inputs of two related tasks. This is then followed by task-specific output layers to perform the corresponding tasks. Note that the two tasks in (Guo et al., 2016) are provided with two different datasets of different text inputs, thereby being similar to the setting we consider in this work. In order to learn the parameters for this model, in each iteration, (Guo et al., 2016) select one of the tasks with some probabilities, sample a mini-batch of examples in the dataset of the chosen task, and update the model parameters using the objective function specific to the chosen task. Consequently, the model parameters for feature representation learning are updated at every iteration while only the model parameters in the output layer for the chosen task are updated at the current iteration.

It has been demonstrated in (Guo et al., 2016) that the alternating method (called *ALT*) is more effective than pre-training the network on a related task and fine-tuning it on the expected task. We thereby consider *ALT* as the baseline for multi-task learning in our work. However, we argue that this baseline is not effective enough to transfer the knowledge from the WSD dataset to ED in our case. This stems from its employment of a single DL model to induce the representations for the text inputs in both tasks. In our case of WSD and ED, although there are some overlap between the semantic differentiation of the two tasks, the labels in the WSD datasets (i.e, the sense ids) tend to be more fine-grained and exhaustive than those in ED. For instance, for the word "*fire*", there might be 12 WSD labels for it in WordNet while the number of possible event types for "*fire*" in the ACE 2005 dataset is only 2 (i.e, "*End-Position*" and "*Attack*"). Eventually, if a single DL model is used to compute the representations for the text inputs in both WSD and ED, the model would suffer from a confusion to distinguish such subtlety in the semantic differentiation.

In order to overcome this issue, we propose to employ two versions M^{wsd} and M^{ed} of the same DL model (with different model parameters) to compute the feature representations for WSD and ED respectively. We then transfer the knowledge from M^{wsd} to M^{ed} by encouraging the representations generated by the two versions M^{wsd} and M^{ed} on the same text inputs to be similar. Formally, let (W^t, p^t, y^t) be an example in the D^{wsd} or D^{ed} dataset ($t \in \{wsd, ed\}$). Also, let R^{wsd} and R^{ed} be the representations for (W^t, p^t) induced by M^{wsd} and M^{ed} respectively:

$$R^{wsd} = M^{wsd}(W^t, p^t), R^{ed} = M^{ed}(W^t, p^t)$$

Such representation vectors are then followed by a task-specific output layer F^t (i.e, feed-forward neural networks followed by a softmax layer) to compute the probability distribution over the possible labels for (W^t, p^t): $P^t(Y^t|R^t) = F^t(R^t)$ where Y^t is the label set for the t task.

If the two models M^{wsd} and M^{ed} were trained separately, the objective function for the t task for the current example would be the negative log-likelihood: $C^t(W^t, p^t, y^t) = -\log P^t(y^t|R^t)$. In this work, instead of just optimizing this objective, we optimize the joint function:

$$C^t(W^t, p^t, y^t) = -\log P^t(y^t|R^t) + \lambda \frac{1}{d_R} \sum_{i=0}^{d_R} \left(R_i^{wsd} - R_i^{ed} \right)^2$$

where λ is a trade-off parameter and d_R is the dimension of the representation vectors.

The second term in the joint objective function enforces that the feature representations learned by M^{wsd} and M^{ed} on the same input context (W^t, p^t) are close to each other ($t \in \{wsd, ed\}$). One the one hand, this representation matching schema helps the two models to communicate to each other so the knowledge from one model can be passed to the other one. On the other hand, the use of two separate models leaves a flexibility for the models to induce the task-specific structures.

Presumably, the objective function (2.2) can simultaneously improve the performance for both tasks of consideration. However, in our case of ED and WSD, it turns out this mechanism actually worsen the performance of the WSD models that were trained separately. We attribute this to the fact that the semantic differentiation in ED is more coarse-grained that that of WSD, causing the ineffectiveness of the datasets for ED to improve WSD performance. Eventually, we will just focus on the ED performance in the experiments.

3 Experiments

3.1 Parameters and Datasets

We use the Semcor dataset (Miller et al., 1994) as the dataset for WSD in this work. This dataset was extracted from the Brown Corpus, and manually annotated with WordNet senses. We evaluate the models on two different datasets for ED:

1. **ACE 2005**: This dataset has 33 event subtypes. We use the same data split with the prior work (Chen et al., 2015; Nguyen and Grishman, 2015b). In particular, 40 newswire documents are used for testing, 30 other documents are reserved for validation, and the 529 remaining documents form the training data.

2. **TAC 2015**: This dataset was released in the Event Nugget Detection Evaluation of the 2015 Text Analysis Conference (TAC) (Mitamura et al., 2015). It comes with 38 event subtypes. We follow the data split in the official evaluation to achieve compatible comparison. As TAC 2015 does not have a development set, we use the best parameters tuned on ACE 2005 for the experiments with TAC 2015.

We use the pre-trained word embeddings provided by (Nguyen and Grishman, 2016d). For *CNN*, *NCNN* and *CNN+BiRNN*, we employ filter sizes of $\{2, 3, 4, 5\}$ with 300 filters for each size as in (Nguyen and Grishman, 2015b), while Gated Recurrent Units (Cho et al., 2014) with 300 hidden units are applied in *BiRNN* and *CNN+BiRNN* (as do (Nguyen and Grishman, 2016d)). For the other parameters, the best values suggested by the development data include: a dropout rate of 0.5, a feed-forward neural network with one hidden layer of 1200 hidden units for the output layers, and the penalty rate λ of 0.01 for both CNN and BiRNN, 0.6 for NCNN, and 0.7 for CNN+BiRNN in the proposed transfer learning method (called *MATCHING*). For simplicity, the same hyper-parameters are used for the two versions of the same network architecture in the *MATCHING* method. We utilize Adadelta (Zeiler, 2012) with back-propagation to train the models in this work.

3.2 Experiments

In this section, we compare the proposed *MATCHING* method with the transfer learning baseline *ALT* in (Guo et al., 2016) and the separate training mechanism for ED (called *SEPARATE*) employed in the previous work for ED (Chen et al., 2015; Nguyen and Grishman, 2015b). Note that in the *SEPARATE* method, the models are only trained on the datasets for ED without utilizing any transfer learning techniques with external datasets. We report the performance when each of the DL methods in Section 2.1 is used as the network to learn the feature representations for ED and WSD.

Tables 1 and 2 present the performance (i.e, F1 scores) of the models on the ACE 2005 and TAC 2015 datasets respectively. The first observation is that the proposed transfer learning method *MATCHING* is consistently better than the baseline method *ALT* across different deep learning models and datasets with large performance gap. This is significantly with $p < 0.05$ and confirms our hypothesis in Section 2.2 about the advantage of the proposed *MATCHING* over the alternating training method *ALT* for ED and WSD. In fact, the performance of the *ALT* method is even worse than the traditional *SEPARATE* method also over different network architectures and datasets. Consequently, training a single deep learning model on a combination of ED and WSD data (as in *ALT*) does not automatically enable the model to learn to exploit the similar structures of the two tasks. In contrast, it hinders the model's ability to effectively extract hidden representations for ED.

Comparing *MATCHING* and *SEPARATE*, we see that *MATCHING* helps to improve *SEPARATE* with respect to difference choices of the DL models. The performance improvement is significant for *CNN* and *BiRNN* on ACE 2005 and for all the models on TAC 2015. Such results demonstrate the effectiveness of the WSD dataset for ED and the ability of the proposed method *MATCHING* to promote knowledge transferring between WSD and ED to improve ED performance.

Regarding the best reported performance, our best performance on ACE (i.e, 71.2% with *CNN*) is comparable with the recent state-of-the-art performance (i.e, Table 1). However, we note that such work heavily relies on the manual annotation of the entity mentions in the documents. Our current work do not employ such information to better reflect the realistic setting. For the TAC 2015 dataset, our best performance is 60.7% with *CNN+BiRNN* although the performance of the other models is also very close. This performance is better than the best performance that has been reported on the TAC 2015 (i.e, Table 2).

Method	CNN	BiRNN	NCNN	CNN+BiRNN
SEPARATE	67.6	67.6	69.3	68.1
ALT	65.1	66.4	65.0	65.2
MATCHING	**71.2**	**69.0**	**69.6**	**68.3**
(Nguyen and Grishman, 2016d)			71.3*	
(Liu et al., 2017)			71.9*	
(Liu et al., 2018)			72.4*	
(Nguyen and Grishman, 2018a)			73.1*	

Table 1: Performance on the ACE 2005 dataset. * indicates the use of entity mention annotation.

Method	CNN	BiRNN	NCNN	CNN+BiRNN
SEPARATE	57.6	59.4	58.3	58.0
ALT	57.6	54.9	48.5	57.5
MATCHING	**60.0**	**60.4**	**60.0**	**60.7**
TAC TOP (Mitamura et al., 2015)			58.4*	
(Nguyen and Grishman, 2018a)			58.8*	

Table 2: Performance on the TAC 2015 dataset. * indicates the use of entity mention annotation.

4 Related Work

Prior works on ED include statistical models with manual feature engineering(Ahn, 2006; Ji and Grishman, 2008; Hong et al., 2011; Li et al., 2013; Venugopal et al., 2014; Li et al., 2015), followed by neural network models, such as CNNs (Nguyen and Grishman, 2015b; Chen et al., 2015; Nguyen et al., 2016b,e; Chen et al., 2017), RNNs (Nguyen et al., 2016a; Jagannatha and Yu, 2016), and attention-based methods (Liu et al., 2017; Nguyen and Nguyen, 2018b).

A similar trend exists in methods proposed for WSD, with feature based methods (Miller et al., 1994; Zhong and Ng, 2010; Taghipour and Ng, 2015) succeeded recently by deep learning methods (Yuan et al., 2016; Raganato et al., 2017).

For multi-task learning in NLP, methods have been proposed for jointly modeling structured prediction tasks (Hatori et al., 2012; Li et al., 2011; Bohnet and Nivre, 2012; Henderson et al., 2013; Lluís et al., 2013; Duong et al., 2015), and for sequence-to-sequence problems (Dong et al., 2015; Luong et al., 2015; Liu et al., 2016; Klerke et al., 2016). The prior work to solve multiple NLP tasks using an unified architecture includes (Collobert and Weston, 2008; Guo et al., 2016).

5 Conclusion

We present a method that improves the performance of deep learning models for ED by training two different versions of the same network architecture for ED and WSD, while encouraging the knowledge transfer between the two versions via representation matching. The proposed method produces better results across a variety of deep learning models.

References

David Ahn. 2006. The stages of event extraction. In *ACL*.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL*.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *ACL*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL-IJCNLP*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL-IJCNLP*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL*.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *ACL*.

Jiang Guo, Wanxiang Che, Haifeng Wang, Ting Liu, and Jun Xu. 2016. A unified architecture for semantic role labeling and relation classification. In *COLING*.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In *ACL*.

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational linguistics*, 39(4):949–998.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *ACL*.

Abhyuday N Jagannatha and Hong Yu. 2016. Bidirectional rnn for medical event detection in electronic health records. In *NAACL*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. *arXiv preprint arXiv:1604.03357*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*.

Xiang Li, Thien Huu Nguyen, Kai Cao, and Ralph Grishman. 2015. Improving event detection with abstract meaning representation. In *Proceedings of ACL-IJCNLP Workshop on Computing News Storylines (CNewS)*.

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for chinese pos tagging and dependency parsing. In *EMNLP*.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018. Event detection via gated multilingual attention mechanism. In *AAAI*.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *ACL*.

Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. In *AAAI*.

Xavier Lluís, Xavier Carreras, and Lluís Màrquez. 2013. Joint arc-factored parsing of syntactic and semantic dependencies. *ACL*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics.

Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2015. Overview of tac kbp 2015 event nugget track. In *TAC*.

Minh Nguyen and Thien Huu Nguyen. 2018b. Who is killed by police: Introducing supervised attention for hierarchical lstms. In *COLING*.

Thien Huu Nguyen, , Adam Meyers, and Ralph Grishman. 2016e. New york university 2016 system for kbp event nugget: A deep learning approach. In *TAC*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016a. Joint event extraction via recurrent neural networks. In *NAACL*.

Thien Huu Nguyen, Lisheng Fu, Kyunghyun Cho, and Ralph Grishman. 2016b. A two-stage approach for extending event detection to new types via neural networks. In *Proceedings of the 1st ACL Workshop on Representation Learning for NLP (RepL4NLP)*.

Thien Huu Nguyen and Ralph Grishman. 2015a. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st NAACL Workshop on Vector Space Modeling for NLP (VSM)*.

Thien Huu Nguyen and Ralph Grishman. 2015b. Event detection and domain adaptation with convolutional neural networks. In *ACL-IJCNLP*.

Thien Huu Nguyen and Ralph Grishman. 2016d. Modeling skip-grams for event detection with convolutional neural networks. In *EMNLP*.

Thien Huu Nguyen and Ralph Grishman. 2018a. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. In *EMNLP*.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*.

Deepak Venugopal, Chen Chen, Vibhav Gogate, and Vincent Ng. 2014. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In *EMNLP*.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. *arXiv preprint arXiv:1603.07012*.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. In *CoRR, abs/1212.5701*.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *ACL*.