Just “OneSeC” for Producing Multilingual Sense-Annotated Data

Bianca Scarlini, Tommaso Pasini and **Roberto Navigli**
Department of Computer Science
Sapienza University of Rome
{scarlini,pasini,navigli}@di.uniroma1.it

Abstract

The well-known problem of knowledge acquisition is one of the biggest issues in Word Sense Disambiguation (WSD), where annotated data are still scarce in English and almost absent in other languages. In this paper we formulate the assumption of *One Sense per Wikipedia Category* and present OneSeC, a language-independent method for the automatic extraction of hundreds of thousands of sentences in which a target word is tagged with its meaning. Our automatically-generated data consistently lead a supervised WSD model to state-of-the-art performance when compared with other automatic and semi-automatic methods. Moreover, our approach outperforms its competitors on multilingual and domain-specific settings, where it beats the existing state of the art on all languages and most domains. All the training data are available for research purposes at http://trainomatic.org/onesec.

1 Introduction

The problem of acquiring knowledge (i.e., the knowledge acquisition bottleneck) is an open issue in Natural Language Processing (NLP). This problem has become even more critical with the advent of deep learning, as a bigger amount of data is needed to meet the requirements of more and more difficult tasks and increasingly complex models. Word Sense Disambiguation (WSD), i.e., the task of associating a word with its meaning in a context (Navigli, 2009), is one of the most affected research areas (Navigli, 2018). The interest in this field has grown remarkably due to the variety of applications that can benefit from it, such as Machine Translation (Neale et al., 2016) or Information Extraction (Delli Bovi et al., 2015). Most approaches to WSD are either supervised or knowledge-based. The former frames the problem as a classification (Zhong and Ng, 2010) or sequence learning (Raganato et al., 2017b) task, in which either a target word or all the content words in a sequence have to be tagged with one of their possible meanings. The latter, instead, exploits graph algorithms on knowledge bases, such as the Personalized PageRank method (Haveliwala, 2002; Agirre et al., 2014), or the densest subgraph heuristic (Moro et al., 2014). Hence, knowledge-based approaches rely on semantic networks such as WordNet[1] (Miller et al., 1990), a manually-curated resource where synonyms are grouped into so-called synsets, or BabelNet[2] (Navigli and Ponzetto, 2010), a large multilingual encyclopedic dictionary that merges together different resources like WordNet, Wikipedia, Wikidata etc. Therefore, in one form or another both approaches to WSD need lexical-semantic data. This is especially crucial in the case of supervised systems, which have proved capable of attaining higher results on English, for which annotated data are available, whereas they fall behind knowledge-based approaches when tested on other languages. Unfortunately, carrying out semantic annotations for a target language requires time, resources and expertise in the field. Thus, in the last few years new approaches have been developed to mitigate the burden of knowledge acquisition by providing automatically or semi-automatically tagged corpora. The main goal of such techniques is to infer the meaning of words occurring in raw sentences by leveraging information drawn from different sources of knowledge, i.e., parallel corpora (Taghipour and Ng, 2015; Delli Bovi et al., 2017), or semantic networks (Pasini and Navigli, 2017; Pasini et al., 2018). Although supervised models achieve competitive results when trained on

[1] https://wordnet.princeton.edu
[2] https://babelnet.org

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 699–709
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

automatically and semi-automatically annotated datasets, a major limitation concerning these approaches is that they are strictly dependent on knowledge sources, which are in their turn difficult to harvest. In fact, on the one hand, parallel corpora require human intervention for translating a collection of texts into one or more different languages. On the other hand, semantic networks rely on manually-annotated lexical-semantic data for enriching the network itself.

In this paper we tackle the knowledge acquisition bottleneck by extending the hypotheses introduced in the two seminal papers by Gale et al. (1992b, One Sense Per Discourse) and Yarowsky (1993, One Sense Per Collocation) to Wikipedia categories, thereby making the following four contributions:

1. We formulate the new assumption of *One Sense per Wikipedia Category*, i.e., all the occurrences of a word across Wikipedia pages in a category share the same word meaning.

2. We propose OneSeC (One Sense per Category), a novel fully-automatic method that produces multilingual sense-annotated datasets on a large scale by mapping Wikipedia categories to word senses.

3. We eliminate the dependency on the structure of a semantic network by relying only on the association between Wikipedia pages and categories and on a sparse vector representation of concepts, i.e., NASARI[3] (Camacho Collados et al., 2016).

4. We prove that OneSeC achieves state-of-the-art results on multilingual WSD and outperforms its automatic and semi-automatic alternatives on English.

2 One Sense Per Category

Preliminaries Wikipedia is the largest electronic encyclopedia, available in approximately 300 languages. It is composed of pages and categories: pages are used to describe named entities and abstract concepts, while categories group pages that convey common semantic information. For example, the *Mouse (computing)* and *Computer keyboard* pages are grouped under the same category, namely, COMPUTING INPUT DEVICES. Similarly, the MONARCHS OF THE UNITED KINGDOM category groups together all the past and present monarchs of the country, e.g. *Elisabeth II*, *Queen Victoria*, etc. Based on this, in what follows we refer to the sentences of a category C as those sentences contained in all the pages of C, and we refer to the occurrences of a lemma in a category C as the occurrences of its inflected forms in the sentences of C.

Automatically annotating Wikipedia Our approach aims at creating a sense-annotated corpus in a target language by leveraging the semantic information contained within Wikipedia categories. Therefore, by relying on the *One Sense per Wikipedia Category* assumption (see Section 1), we infer the meaning of words occurring in Wikipedia sentences by exploiting the information provided by their categories. For example, the lemma[4] *spring#n* appears in more than 8K categories, including SEASONS and MECHANICS. At the end of our procedure, OneSeC automatically assigns the *metal elastic device* sense to all the occurrences of *spring#n* in MECHANICS and the *season* sense to those in SEASONS.

Given the whole Wikipedia together with its associations between pages and categories and given a lexicon of words $\mathcal{L}$, our approach computes a semantically-tagged dataset – where words in $\mathcal{L}$ are annotated with their correct meaning – by performing the following three steps:

- **Category Representation**, which represents a lexeme-category pair (l, C) as the Bag Of Words of the sentences of the category C in which the lemma l appears (Section 2.1).

- **Sense Assignment**, which assigns a sense s of the lemma l to each lexeme-category pair (l, C) (Section 2.2).

- **Sentence Sampling**, which extracts a certain number of sentences for each sense s of each lemma l in the lexicon $\mathcal{L}$ by exploiting the association between lexeme-category pairs and word senses computed in the previous step (Section 2.3).

2.1 Category Representation

The first step aims at representing each lexeme-category pair (l, C) with a Bag Of Words (BOW). To that end, we lemmatise and POS tag the text of

[3] http://lcl.uniroma1.it/nasari/

[4] We use lemma and lexeme, i.e., a lemma#pos, interchangeably.

BOW	mouse, cat, animal, vehicle, rodent, mice, mammal
Mouse (animal)	mouse, animal, rodent, mice, mammal, cat
Mouse (device)	mouse, computer, keyboard, device, input, output

Table 1: Excerpt of the sorted components of an example category's BOW (first line) and two NASARI vectors (second and third line).

(*spring#n*, Seasons)		(*match#n*, Sports law)	
The season of growth	0.63	A formal contest	0.49
Natural flow of water	0.10	Score needed to win	0.21
Movement upwards	0.08	Exact duplicate	0.07

Table 2: Excerpt of the sense distribution of spring#n and match#n for one of their categories.

Mouse (Animal)	Score	Mouse (Device)	Score
Mice	2.91	Computing input devices	3.35
Invasive mammal species	2.91	Pointing devices	3.24
Rodents	2.82	Computer connectors	3.24
Rodents of Australia	2.70	Personal computers	3.07
Rodents of Africa	2.65	Computer keyboards	2.87

Table 3: Excerpt of the most related categories for the device and animal senses of mouse.

each page in C and retain only the content words in each sentence. Then, we consider all the sentences of C in which l appears at least once and we count the frequency of each other lemma occurring in the selected sentences. Finally, we build the BOW of (l, C) in which each dimension corresponds to a lemma that is associated with its frequency, thus giving greater importance to more frequent words. For example, the pair (*spring#n*, Mechanics) contains words such as *force* and *gravity*, while the pair (*match#n*, Sports law) includes *team* and *play*.

2.2 Sense Assignment

The second step aims at assigning a sense distribution to each lexeme-category pair. We exploit the BOW we computed and the NASARI lexical vectors (Camacho Collados et al., 2016) to represent categories and synsets, respectively. NASARI leverages Wikipedia pages to provide a sparse representation of BabelNet synsets, having words as their dimensions weighted by their lexical specificity (Lafon, 1980). NASARI has been used to compute the semantic similarity between two concepts (Pilevar et al., 2013) in combination with the Weighted Overlap (WO), which has proven to work better than cosine similarity for comparing sparse vectors. It takes as input two vectors v_1 and v_2 and computes their similarity by considering the ranks of the components shared by both vectors[5]. However, as it takes into account only the common dimensions, it also gives a high similarity value when the two vectors share just a few dimensions with similar rankings. In light of this, we modified the original formula and added a weight factor Ψ as follows:

$$WO(v_1, v_2) = \Psi \frac{\sum_{w \in \mathcal{O}} (r_w^{v_1} + r_w^{v_2})^{-1}}{\sum_{i=1}^{|\mathcal{O}|} (2i)^{-1}} \tag{1}$$

where $\mathcal{O}$ is the intersection set between the dimensions of v_1 and v_2, $r_w^{v_i}$ is the rank of the dimension corresponding to the word w in the vector v_i and Ψ is a logarithmic function that depends on the size of $\mathcal{O}$ and is defined as $\Psi = ln(|\mathcal{O}| + 1)$.

For example, given the BOW for a category related to the animal mouse and the two NASARI vectors for the *animal* and *device* senses of *mouse* as in Table 1, the standard weighted overlap scores the *animal* sense 0.93 and the *device sense* 1.00, even though the latter has only the first dimension in common. When we add the logarithmic factor Ψ, instead, the first sense is scored 1.80 while the second is scored 0.69.

Therefore, for each lexeme-category pair (l, C) we compute the WO between B_C, i.e., the BOW representation of the category C (see Section 2.1), and each NASARI vector associated with a given sense of l. Thus, given a set of weighted overlap scores $\{WO(B_C, s_1), \dots, WO(B_C, s_n)\}$, where $s_1 \dots s_n$ are the senses of l, we assign to (l, C) the sense that maximises the similarity with the category BOW as follows:

$$sense(l, C) = \arg\max_{s_i} \{WO(B_C, s_i)\}$$

In Table 2 we show the distribution of senses for one category of *spring#n* and *match#n*, respectively. As one can see, given the pair (*spring#n*, Seasons) we select the *season* sense of *spring#n* as it is the highest ranked one in terms of WO, while the *formal contest* meaning of *match#n* is selected for (*match#n*, Sports law).

2.3 Sentence Sampling

Once each lexeme-category pair (l, C) is associated with one sense, we can reverse the relation having – for each sense of l – a list of categories $C_1, \dots, C_m$ sorted by weighted overlap. For example, in Table 3 we show an excerpt of the most

[5] We note that the components of each vector are ranked according to their weights.

related categories for the *animal* and the *device* meanings of the lemma *mouse#n*. As one can see, the *animal* sense is mostly related to categories that concern the animal world, e.g. MICE, RODENTS, etc., while the *device* sense to the electronic device world, e.g. COMPUTING INPUT DEVICES, POINTING DEVICES, etc. Therefore, for each sense s_i of l we sample a set of $\mathcal{K}_{s_i}$ sentences from $C_1, \ldots, C_{m_i}$ that depends on the BabelNet ordering of senses. Following Pasini and Navigli (2017) we compute $\mathcal{K}_{s_i}$ applying a Zipfian distribution:

$$\mathcal{K}_{s_i} = \frac{\mathcal{K}}{i^z} \tag{2}$$

where $\mathcal{K}$ and z are two system parameters that define, respectively, the number of examples to assign to the first sense of a lemma and how fast the function decreases, while i is the sense position in BabelNet. In the case that we find only β sentences for the first sense of l, with $\beta < \mathcal{K}_{s_1}$, we scale down all $\mathcal{K}_{s_i}$ by setting $\mathcal{K} = \beta$, i.e., we consider the maximum number of examples as those that are actually available for the first sense. For example, if we have $z = 2.0$ and $\mathcal{K} = 500$ but we can retrieve only 100 sentences for the sense s_1, we set $\mathcal{K} = 100$ when computing $\mathcal{K}_{s_i}$ for $i > 1$. Hence, the number of sentences to be associated with s_2 is 25, rather than 125, thus maintaining the distribution across senses balanced.

In order to provide different contexts of use for a given sense s_i, we sample $\mathcal{K}_{s_i}^{C_j}$ sentences from each category C_j. $\mathcal{K}_{s_i}^{C_j}$ is computed as follows:

$$\mathcal{K}_{s_i}^{C_j} = \mathcal{K}_{s_i} \frac{j^{-1}}{\sum\limits_{j'=1}^{m_i} j'^{-1}} \tag{3}$$

where the second term is a smoothed version of the category rank reciprocal[6], i.e., it is normalised by the sum of the reciprocal of each category rank (from 1 to m_i).

Once we have determined the number of examples to draw from each category, we sample the sentences according to their perplexity, which we compute with a Neural Language Model trained on WikiText103 (Howard and Ruder, 2018)[7].

The result of the above three steps is a semantically-annotated corpus where each meaning s of each lemma $l \in \mathcal{L}$ is associated with a set of sentences in which l is tagged with s.

3 Experimental Setup

We exploited the Word Sense Disambiguation task to assess the quality of our automatically-generated corpus. Therefore, we trained a reference WSD model on the data generated by OneSeC and compared the results against those achieved by the same model trained on other resources.

In what follows we introduce the reference Word Sense Disambiguation system, the test bed, the comparison systems and how we tuned the two parameters $\mathcal{K}$ and z.

Reference system We carried out the evaluation with two different WSD models: the SVM-based system It Makes Sense (Zhong and Ng, 2010, IMS) and the Bi-LSTM-based model introduced by Raganato et al. (2017b). For the latter we used MUSE embeddings (Lample et al., 2018) in the input layer, a learning rate of 0.5 and followed Raganato et al. (2017b) for all the other hyperparameters. Depending on the setting, English or multilingual, we chose the best-performing system on a development set: Senseval-2 for English and an in-house development set for all the other languages[8]. For both models, unless differently stated, we used the Most Frequent Sense (MFS) of a lemma, i.e., its first-ranked meaning in BabelNet, as backoff strategy when the system was not able to provide an answer.

Test bed We used the evaluation framework for English all-words WSD made available by Raganato et al. (2017a). This comprises all the past test sets, including Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), SemEval-2007 (Pradhan et al., 2007), SemEval-2013 (Navigli et al., 2013), SemEval-2015 (Moro and Navigli, 2015) and ALL, i.e., the concatenation of all the aforementioned datasets. For the multilingual evaluation, instead, we used the all-words multilingual WSD tasks of SemEval-2013 (Navigli et al., 2013) and SemEval-2015 (Moro and Navigli, 2015). For both settings, we focused on nouns only, as NASARI vectors are available mainly for nominal concepts.

Following the literature, we report the F1 measure on all the test sets unless stated differently.

[6] Recall that the categories associated with the sense s are sorted by weighted overlap.

[7] http://files.fast.ai/models/wt103/

[8] The development set of each language comprises 50 manually-annotated word-sense pairs.

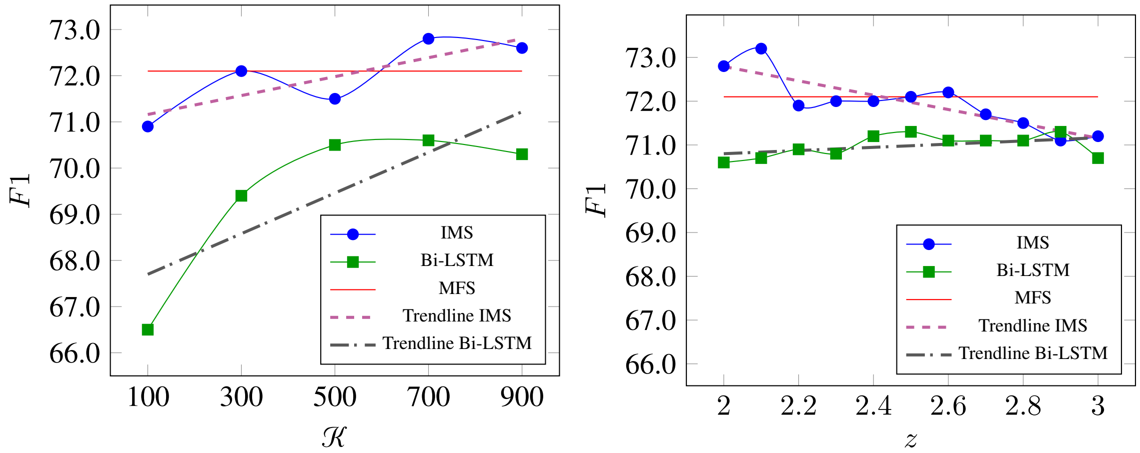

Figure 1: Performance on the development set of IMS and the Bi-LSTM model trained on OneSeC when $z = 2.0$ and $\mathcal{K}$ ranges between 100 and 900 (left) and when $\mathcal{K} = 700$ and z ranges between 2.0 and 3.0 (right).

English parameter tuning We tune the parameters $\mathcal{K}$ and z introduced in Section 2.3 so as to maximise the performance of the reference system on the development set. We used Senseval-2 as tuning corpus and varied $\mathcal{K}$ between 100 and 900 with a 200 step and z between 2.0 and 3.0 with a 0.1 step. We ran both models, IMS and Bi-LSTM, for each parameter value and chose the one that performed best. In Figure 1 (left) we show the results of the two systems when trained on OneSeC where z is set to 2.0 and $\mathcal{K}$ is increased from 100 to 900. As can be seen, the Bi-LSTM trend increases more rapidly than the IMS one. However, its results are always lower than those attained by its alternative. IMS, in fact, scores almost 5 points higher starting from $\mathcal{K} = 100$ and maintains its lead through all the values of $\mathcal{K}$. It reaches a plateau when $\mathcal{K} = 700$, which we interpret as the *plateau of knowledge*. Indeed, increasing the number of examples degrades IMS performance as no more informative sentences are found for a given sense. Once $\mathcal{K}$ was set to 700 both for IMS and Bi-LSTM, we ran the same experiment varying z. As one can see in Figure 1 (right), IMS achieves the highest score when $z = 2.1$ while Bi-LSTM when $z = 2.9$. While IMS seems sensitive to this parameter, attaining better performance when the distribution of classes in training is more balanced, the neural model trend is almost constant, indicating it is less dependent on the sense distribution.

Therefore, we chose IMS as our WSD reference system as it consistently outperformed its neural-network alternative. In the following we report the results of IMS trained on OneSeC with $\mathcal{K} = 700$ and $z = 2.1$.

Multilingual parameter tuning We varied $\mathcal{K}$ and z as for English and computed the performance separately on each language-specific development dataset. We then chose the parameters leading the reference model to the highest results averaged across all languages. Contrary to what was the case for English, the Bi-LSTM model outperformed IMS on most of the settings and achieved the highest score with $\mathcal{K} = 200$ and $z = 2.0$. Hence, we report multilingual results attained by the Bi-LSTM model when trained on OneSeC with $\mathcal{K} = 200$ and $z = 2.0$.

Comparison systems We compared OneSeC with a manual, a semi-automatic and a fully-automatic alternative:

- **SemCor** (Miller et al., 1993): the most used training corpus in WSD, which provides more than $200K$ manual annotations.

- **OMSTI** (Taghipour and Ng, 2015): a semi-automatic approach that extracts semantically-annotated data by exploiting parallel data to reduce the ambiguity of the target language. Since the resource contains SemCor by default, we considered only the semi-automatically generated examples in order to guarantee a fair comparison with OneSeC.

- **Train-O-Matic**[9] (Pasini and Navigli, 2017, TOM): a knowledge-based method for the automatic generation of sense-annotated data.

[9] http://trainomatic.org

Dataset	OneSeC			OMSTI			Train-O-Matic			SemCor		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Senseval-2	72.3	69.1	**70.7**	64.8	38.5	39.6	69.5	65.5	67.4	73.5	61.3	66.8
Senseval-3	66.5	62.1	64.2	55.7	31.0	39.8	66.1	63.1	64.6	73.2	67.6	**70.2**
SemEval-07	63.7	62.9	63.3	64.1	35.9	46.0	59.8	59.8	59.8	68.9	65.4	**67.1**
SemEval-13	64.0	58.3	**61.0**	50.7	23.4	32.0	61.3	53.3	57.0	63.2	55.4	59.0
SemEval-15	69.2	64.8	**66.9**	57.0	26.7	36.4	67.0	62.3	64.6	65.3	56.3	60.5
ALL	67.3	62.8	**64.9**	56.5	27.0	36.5	65.1	59.7	62.3	68.3	59.9	63.8

Table 4: Performance of IMS trained on different corpora on the English all-words WSD tasks when the MFS is disabled.

Dataset	OneSeC	TOM	OMSTI	SemCor	MFS
Senseval-2	73.2	70.5	74.1	**76.8**	72.1
Senseval-3	68.2	67.4	67.2	**73.8**	72.0
SemEval-07	63.5	59.8	62.3	**67.3**	65.4
SemEval-13	**66.5**	65.5	62.8	65.5	63.0
SemEval-15	**70.8**	68.6	63.1	66.1	66.3
ALL	69.0	67.3†	66.4†	**70.4**	67.6

Table 5: Results of IMS trained on different corpora on the English all-words WSD tasks. † marks statistical significance between OneSeC and its competitors.

Dataset	OneSeC	TOM	OMSTI	Total
Senseval-2	401	400	197	436
Senseval-3	424	435	197	469
SemEval-07	125	127	68	127
SemEval-13	656	629	249	751
SemEval-15	228	226	102	253
ALL	1359	1350	456	1557

Table 6: Number of nominal lemmas covered by each corpus.

For the multilingual setting, instead, due to the lack of manually sense-annotated data for non-English languages, we compared directly OneSeC against the best participating system in each task and Train-O-Matic. To set a level playing field, we also report the results attained by the Bi-LSTM model when trained on Train-O-Matic corpora for the tested languages.

4 Results

4.1 English All-Words WSD

We proceed by testing the reference WSD system on the data provided by OneSeC, Train-O-Matic, OMSTI and SemCor on the English all-words tasks.

In Table 5 we compare the results of IMS when trained on different corpora. As one can see, OneSeC achieves the best results on ALL when compared to automatic and semi-automatic approaches, and ranks second only with respect to SemCor. Interestingly enough, OneSeC beats its manual competitor on SemEval-2013 by 1 point and on SemEval-2015 by 4.7 points, an impressive result considering that OneSeC does not involve any human intervention during the generation of the corpus. In Table 5 we also report the statistical significance between OneSeC and its competitors on the ALL dataset by juxtaposing a † symbol next to the score. In order to do this, we computed the McNemar's χ^2 test (McNemar, 1947) with significance level $\alpha = 0.01$ between OneSeC and SemCor. It resulted in no statistical significance, meaning that IMS trained on OneSeC is in the same ballpark as when trained on SemCor. We note that the goal of this work was not to achieve state-of-the-art results on English WSD compared to manually-annotated corpora. However, performing competitively on standard benchmarks represents one step further towards getting rid of the limitation imposed by resources like SemCor. Moreover, our approach outperforms Train-O-Matic, our direct competitor, on all the datasets, with the highest increment of 3.7 points on SemEval-2007, while scoring almost 2 points higher than TOM overall.

OneSeC also attains higher results when compared with a semi-automatic approach like OMSTI. In fact, OMSTI is surpassed on all the datasets but Senseval-2 and scores 2.6 F1 points less on the ALL dataset. This is *per se* a remarkable result as OneSeC is automatic, while OMSTI relies on parallel corpora and manual effort to align senses across languages. Furthermore, we show that OneSeC results are statistically significant in comparison to those attained by TOM and OMSTI. We also note that, similarly to TOM, OneSeC covers almost all the lemmas in each test set (see Table 6), while OMSTI is able to provide

Dataset	Domain	Size	Backoff	OneSeC			TOM			OMSTI			MFS
				P	R	F1	P	R	F1	P	R	F1	
SemEval-13	Biology	135	MFS	67.4	67.4	**67.4**	63.0	63.0	63.0	65.9	65.9	65.9	64.4
			-	65.1	60.7	**62.8**	59.0	53.3	56.0	48.1	18.5	26.7	
	Climate	194	MFS	68.0	68.0	68.0	68.1	68.1	**68.1**	68.0	68.0	68.0	67.5
			-	65.0	54.6	**59.4**	63.4	50.0	55.9	58.0	24.2	34.2	
	Finance	219	MFS	69.4	69.4	**69.4**	68.0	68.0	68.0	64.4	64.4	64.4	56.2
			-	67.2	61.6	**64.3**	62.1	51.6	56.4	57.4	28.3	37.9	
	Health Care	138	MFS	65.9	65.9	**65.9**	65.2	65.2	65.2	52.9	52.9	52.9	56.5
			-	62.4	56.5	**59.3**	61.3	55.1	58.0	34.6	18.4	24.0	
	Politics	279	MFS	68.8	68.8	**68.8**	65.2	65.2	65.2	63.4	63.4	63.4	67.7
			-	67.0	63.4	**65.2**	62.5	54.8	58.4	54.1	21.5	30.8	
	Social Issues	349	MFS	66.5	66.5	66.5	68.5	68.5	**68.5**	65.6	65.6	65.6	67.6
			-	62.5	55.0	**58.5**	63.1	53.0	57.6	54.7	25.2	34.5	
	Sport	330	MFS	61.8	61.8	**61.8**	60.3	60.3	60.3	58.8	58.8	58.8	57.6
			-	61.8	57.3	**58.8**	58.3	54.6	56.4	45.0	23.0	30.4	
SemEval-15	Biomedicine	100	MFS	78.4	78.4	**78.4**	76.3	76.3	76.3	64.9	64.9	64.9	71.1
			-	77.8	72.2	**74.9**	76.1	72.2	74.1	60.5	26.8	37.2	
	Maths & Pc	97	MFS	60.0	60.0	**60.0**	50.0	50.0	50.0	36.0	36.0	36.0	40.9
			-	59.8	58.0	**58.9**	50.0	47.0	48.5	21.2	11.0	14.5	

Table 7: Domain-specific evaluation on SemEval-2013 and SemEval-2015 of IMS trained on OneSeC, TOM and OMSTI.

annotated examples for only half of the instances. Therefore, IMS – when trained on OMSTI – resorts heavily to the MFS backoff strategy.

In light of this, we computed precision (P), recall (R) and their harmonic mean (F1) when no backoff strategy was used, as shown in Table 4. As one can see, OMSTI's performance drops heavily by roughly 30 points, confirming the figures in Table 6. Train-O-Matic's results, in contrast, remain consistent, scoring 1.5 F1 points less than SemCor overall and managing to beat it on 2 datasets. OneSeC, instead, leads IMS to the highest results overall, managing to surpass those achieved, not only by its direct competitors, but also by SemCor.

The results attest the high quality of our corpus, hence crowning OneSeC as the best choice over its competitors and even over manually-curated corpora when no back-off strategy is available.

4.2 Augmenting SemCor

To further investigate the quality of the examples provided by OneSeC, we augmented SemCor with our automatically-tagged sentences (SemCor+OneSeC). We added examples to SemCor in two cases:

1. When a word in OneSeC lexicon never appears tagged in SemCor.

2. When not all senses of a word are covered by at least one example in SemCor.

In the first case we provided annotated sentences for all the senses of the target word with $\mathcal{K} = 700$ and $z = 2.1$. In the second case, instead, we generated examples only for those senses s_i of a word w that are missing in SemCor. We determined the number of examples for s_i by following the Zipfian distribution in Formula 2 with $z = 2.1$ and $\mathcal{K} = |examples(s_1, w)|$, i.e., the number of examples in SemCor where w occurs tagged with its most frequent sense s_1. SemCor+OneSeC achieves 70.7 F1 points on ALL, beating SemCor alone (70.4) and SemCor+OMSTI (70.5)[10].

4.3 Domain-Specific Evaluation

In Table 7 we show the results achieved by IMS on each specific domain of SemEval-2013 and SemEval-2015. As shown in the two tables, when compared with TOM and OMSTI, OneSeC leads IMS to consistently outperform all the other approaches on SemEval-2015 and most of the domains of SemEval-2013. In fact, OneSeC scores lower only in 2 out of the 7 SemEval-2013 domains, whereas Train-O-Matic, instead, scores 0.1 and 2 points higher. However, when the MFS is disabled (second row of each domain), OneSeC is the best system across the board, demonstrating it can also provide valuable examples for those words that are specific to a domain.

[10] We refer to SemCor+OMSTI as the dataset containing all tagged sentences of both SemCor and OMSTI corpora.

4.4 Multilingual All-Words WSD

Finally, we move our focus to testing the ability of OneSeC to scale to different languages. In Tables 8 and 9 we show the results obtained by Bi-LSTM trained on OneSeC and Train-O-Matic (TOM - Bi-LSTM) when the MFS backoff strategy is disabled. We compare the aforementioned approaches with the best participating system in SemEval-2013 and SemEval-2015, i.e., UMCC-DLSI's (Gutiérrez Vázquez et al., 2010) best run for the Spanish test set of SemEval-2013 and IMS trained on Train-O-Matic for all other datasets (Pasini et al., 2018). OneSeC proved, once again, to be the best system across the board, achieving state-of-the-art results on all languages. Our approach outperforms its competitors on all datasets, with the highest increment of 7.4 points on the French test set for SemEval-2013, while scoring on average 3.2 F1 points higher compared to the existing state of the art.

Results show that OneSeC is a robust approach that is able to scale across languages and domains. It goes beyond the findings of Train-O-Matic and raises the state-of-the-art bar in multilingual WSD.

Lang	OneSeC			TOM - Bi-LSTM			Best
	P	R	F1	P	R	F1	F1
IT	72.3	64.5	**68.2**	65.4	60.6	62.9	68.0 ⋄
ES	76.0	68.3	**72.0**	71.7	66.8	69.2	71.0 ∗
FR	79.2	70.9	**74.8**	71.0	64.2	67.4	61.0 ⋄∗
DE	83.0	68.5	**75.1**	77.5	64.1	70.2	63.0 ⋄

Table 8: Comparison of Bi-LSTM trained on OneSeC and TOM with the best system (Best) on SemEval-2013. ⋄ Train-O-Matic, ∗ UMCC-DLSI.

Lang	OneSeC			TOM - Bi-LSTM			Best
	P	R	F1	P	R	F1	F1
IT	65.0	60.2	**62.5**	61.6	58.3	59.9	59.9 ⋄
ES	67.8	58.4	**62.8**	62.5	56.1	59.2	57.9 ⋄

Table 9: Comparison of Bi-LSTM trained on OneSeC and TOM with the best system (Best) on SemEval-2015. ⋄ Train-O-Matic, ∗ UMCC-DLSI.

5 Related Work

Word Sense Disambiguation is a well-established task in the field of Natural Language Processing and it has been tackled from many different angles over the past years. One of the major problems concerning WSD has been the so-called knowledge acquisition bottleneck (Gale et al., 1992a), i.e., the paucity of lexical-semantic data. In fact, semantic resources are mainly exploited by WSD models in one of two different ways: as structured knowledge to identify the meaning of a word in a context in knowledge-based models (Moro et al., 2014; Agirre et al., 2014; Chaplot and Salakhutdinov, 2018), and as training data to fit the parameters of a classifier in supervised models (Zhong and Ng, 2010; Yuan et al., 2016; Raganato et al., 2017b; Luo et al., 2018).

On the one hand, knowledge-based models have proved to be more versatile when it comes to disambiguating less frequent words and texts in low-resourced languages, even though they suffer from the lack of statistical evidence of lexical context. On the other hand, supervised models have consistently attained higher results in English WSD (Raganato et al., 2017a), however at the cost of less flexibility and lower results when scaling to other languages (Raganato et al., 2017b). Thus, research has recently been focused on new techniques that aim at mitigating the effects of the knowledge-acquisition bottleneck by automatically creating high-quality, sense-annotated training corpora. Some earlier attempts consisted of annotating examples from the Web by exploiting the target words' monosemous relatives (Agirre and Martínez, 2004). But a major drawback of this kind of approach is its limited coverage. In fact, a training example can be provided only for those senses with at least one monosemous related concept. Raganato et al. (2016) presented in their paper a method for the automatic construction of a Semantically Enriched Wikipedia (SEW), where the number of hyperlink annotations was enlarged by means of a set of heuristics. As an outcome they released a corpus containing more than 200 million annotations for approximately 4 million concepts and named entities. Another approach was developed by Otegi et al. (2016) to enrich the multilingual text of Europarl (Koehn, 2005) and QTLeap (Agirre et al., 2014) with several features, including semantic annotations in 6 different languages. Parallel corpora were exploited also in the more recent work of Taghipour and Ng (2015, OMSTI)[11], who presented a semi-automatic approach that creates a novel semantically-annotated dataset by leveraging the manual effort made to align senses across different languages.

In contrast, recent methods have been able to fully automatise the whole process while simulta-

[11] http://lcl.uniroma1.it/wsdeval/training-data

neously producing high-quality resources. For example, Delli Bovi et al. (2017) exploited an external WSD system, i.e., Babelfy (Moro et al., 2014), and the richer context provided by aligned sentences, to carry out semantic annotations for Europarl. Instead, Pasini and Navigli completely removed the need for parallel corpora (Pasini and Navigli, 2017; Pasini et al., 2018) and for the WordNet backoff strategy (Pasini and Navigli, 2018) by introducing Train-O-Matic and two automatic methods for inducing the sense distribution.

Our work follows this latter line of research and, similarly to the aforementioned approaches, automatically provides multilingual sense-annotated data on a large scale. OneSeC stands out from its alternatives as it does not depend either on the structure of a semantic network (like Train-O-Matic), or on external WSD models (like EuroSense). In our approach, in fact, we only rely on Wikipedia categories and NASARI vectors to inject semantic information at sentence level.

6 Conclusions

In this paper we presented OneSeC, a novel method for the automatic creation of multilingual sense-annotated corpora on a large scale. Our approach relieves the burden of human intervention, hence mitigating the knowledge acquisition bottleneck besetting WSD training data. Moreover, we take a further step towards removing any dependency on a semantic-network structure by exploiting only Wikipedia categories and a sparse vector representation of concepts for creating our datasets. OneSeC outperforms its automatic and semi-automatic alternatives on the English WSD task, and achieves results in the same ballpark as those attained when manually-curated corpora are used for training. Furthermore, OneSeC scales to multiple languages without any additional human effort. Indeed, our approach also proved to be capable of producing high-quality training data for low-resourced languages, leading a WSD supervised model to achieve state-of-the-art results on all the datasets of the multilingual WSD tasks. We release more than one million tagged sentences for English, Spanish, Italian, French and German at `http://trainomatic.org/onesec`.

As future work we plan to exploit a subset of the Wikipedia categories as coarse-grained sense inventory and enrich our dataset with coarser labels, hence enabling WSD at different granularities.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487 under the European Union's Horizon 2020 research and innovation programme.

References

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random Walks for Knowledge-Based Word Sense Disambiguation. *Computational Linguistics*, 40(1):57–84.

Eneko Agirre and David Martínez. 2004. Unsupervised WSD Based on Automatically Retrieved Examples: The Importance of Bias. In *Proc. of EMNLP*, pages 25–33.

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

Devendra Singh Chaplot and Ruslan Salakhutdinov. 2018. Knowledge-based word sense disambiguation using topic models. In *Proc. of AAAI*, pages 5062–5069.

Claudio Delli Bovi, José Camacho Collados, Alessandro Raganato, and Roberto Navigli. 2017. EuroSense: Automatic Harvesting of Multilingual Sense Annotations from Parallel Text. In *Proc. of ACL*, pages 594–600.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-Scale Information Extraction from Textual Definitions through Deep Syntactic and Semantic Analysis. *Transactions of ACL*, 3:529–543.

Philip Edmonds and Scott Cotton. 2001. Senseval-2: Overview. In *Proc. of SENSEVAL*, pages 1–5.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992a. A method for disambiguating word senses in a corpus. *Computers and the Humanities*, 26:415–439.

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

Yoan Gutiérrez Vázquez, Antonio Fernandez Orquín, Andrés Montoyo Guijarro, and Sonia Vázquez Pérez. 2010. UMCC-DLSI: Integrative resource for disambiguation task. In *Proc. of the 5th International Workshop on Semantic Evaluation*, pages 427–432.

Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proc. of the 11th international conference on World Wide Web*, pages 517–526. ACM.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proc. of ACL*, pages 328–339.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of MT summit*, pages 79–86.

Pierre Lafon. 1980. Sur la variabilité de la fréquence des formes dans un corpus. *Mots*, 1(1):127–165.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. *International Conference on Learning Representations*, pages 1–14.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating Glosses into Neural Word Sense Disambiguation. In *Proc. of ACL*, pages 2473–2482.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, pages 153–157.

George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proc. of the Workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proc. of SemEval-2015*, pages 288–297.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transaction of ACL*, 2:231–244.

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

Roberto Navigli. 2018. Natural language understanding: Instructions for (present and future) use. In *Proc. of IJCAI*, pages 5697–5702.

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

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proc. of ACL*, pages 216–225.

Steven Neale, Luís Gomes, Eneko Agirre, Oier Lopez de Lacalle, and António Branco. 2016. Word Sense-Aware Machine Translation: Including Senses as Contextual Features for Improved Translation Models. In *Proc. of LREC*, pages 2777–2783.

Arantxa Otegi, Nora Aranberri, Antonio Branco, Jan Hajic, Steven Neale, Petya Osenova, Rita Pereira, Martin Popel, Joao Silva, Kiril Simov, and Eneko Agirre. 2016. QTLeap WSD/NED Corpora: Semantic Annotation of Parallel Corpora in Six Languages. In *Proc. of LREC*, pages 3023–3030.

Tommaso Pasini, Francesco Elia, and Roberto Navigli. 2018. Huge Automatically Extracted Training Sets for Multilingual Word Sense Disambiguation. In *Proc. of LREC*, pages 1694–1698.

Tommaso Pasini and Roberto Navigli. 2017. Train-O-Matic: Large-Scale Supervised Word Sense Disambiguation in Multiple Languages without Manual Training Data. In *Proc. of EMNLP*, pages 78–88.

Tommaso Pasini and Roberto Navigli. 2018. Two knowledge-based methods for high-performance sense distribution learning. In *Proc. of AAAI*, pages 5374–5381.

Mohammad Taher Pilevar, David Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In *Proc. of ACL*, pages 1341–1351.

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 Task 17: English Lexical Sample, SRL and All Words. In *Proc. of SemEval-2007*, pages 87–92.

Alessandro Raganato, José Camacho Collados, and Roberto Navigli. 2017a. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proc. of EACL*, pages 99–110.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2016. Automatic Construction and Evaluation of a Large Semantically Enriched Wikipedia. In *Proc. of IJCAI*, pages 2894–2900.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural Sequence Learning Models for Word Sense Disambiguation. In *Proc. of EMNLP*, pages 1167–1178.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proc. of SENSEVAL-3*, pages 41–43.

Kaveh Taghipour and Hwee Tou Ng. 2015. One Million Sense-Tagged Instances for Word Sense Disambiguation and Induction. In *Proc. of CoNLL*, pages 338–344.

David Yarowsky. 1993. One sense per collocation. In *Proc. of the workshop on Human Language Technology*, pages 266–271.

Dayu Yuan, Ryan Doherty, Julian Richardson, Colin Evans, and Eric Altendorf. 2016. Semi-supervised Word Sense Disambiguation with Neural Models. In *Proc. of COLING Technical Papers*, pages 1374–1385.

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