Train-O-Matic: Large-Scale Supervised Word Sense Disambiguation in Multiple Languages without Manual Training Data

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

Abstract

Annotating large numbers of sentences with senses is the heaviest requirement of current Word Sense Disambiguation. We present Train-O-Matic, a language-independent method for generating millions of sense-annotated training instances for virtually all meanings of words in a language's vocabulary. The approach is fully automatic: no human intervention is required and the only type of human knowledge used is a WordNet-like resource. Train-O-Matic achieves consistently state-of-the-art performance across gold standard datasets and languages, while at the same time removing the burden of manual annotation. All the training data is available for research purposes at http://trainomatic.org.

1 Introduction

Word Sense Disambiguation (WSD) is a key task in computational lexical semantics, inasmuch as it addresses the lexical ambiguity of text by making explicit the meaning of words occurring in a given context (Navigli, 2009). Anyone who has struggled with frustratingly unintelligible translations from an automatic system, or with the meaning bias of search engines, can understand the importance for an intelligent system to go beyond the surface appearance of text.

There are two mainstream lines of research in WSD: supervised and knowledge-based WSD. Supervised WSD frames the problem as a classical machine learning task in which, first a training phase occurs aimed at learning a classification model from sentences annotated with word senses and, second the model is applied to previously-unseen sentences focused on a target word. A key difference from many other problems, however, is that the classes to choose from (i.e., the senses of a target word) vary for each word, therefore requiring a separate training process to be performed on a word by word basis. As a result, hundreds of training instances are needed for each ambiguous word in the vocabulary. This would necessitate a million-item training set to be manually created for each language of interest, an endeavour that is currently beyond reach even in resource-rich languages like English.

The second paradigm, i.e., knowledge-based WSD, takes a radically different approach: the idea is to exploit a general-purpose knowledge resource like WordNet (Fellbaum, 1998) to develop an algorithm which can take advantage of the structural and lexical-semantic information in the resource to choose among the possible senses of a target word occurring in context. For example, a PageRank-based algorithm can be developed to determine the probability of a given sense being reached starting from the senses of its context words. Recent approaches of this kind have been shown to obtain competitive results (Agirre et al., 2014; Moro et al., 2014). However, due to its inherent nature, knowledge-based WSD tends to adopt bag-of-word approaches which do not exploit the local lexical context of a target word, including function and collocation words, which limits this approach in some cases.

In this paper we get the best of both worlds and present Train-O-Matic, a novel method for generating huge high-quality training sets for all the words in a language's vocabulary. The approach is language-independent, thanks to its use of a multilingual knowledge resource, BabelNet (Navigli and Ponzetto, 2012), and it can be applied to any kind of corpus. The training sets produced with Train-O-Matic are shown to provide competitive performance with those of manually and semi-

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 78–88
Copenhagen, Denmark, September 7–11, 2017.

automatically tagged corpora. Moreover, state-of-the-art performance is also reported for low resourced languages (i.e., Italian and Spanish) and domains, where manual training data is not available.

2 Building a Training Set from Scratch

In this Section we present Train-O-Matic, a language-independent approach to the automatic construction of a sense-tagged training set. Train-O-Matic takes as input a corpus C (e.g., Wikipedia) and a semantic network $G = (V, E)$. We assume a WordNet-like structure of G, i.e., V is the set of concepts (i.e., synsets) such that, for each word w in the vocabulary, $Senses(w)$ is the set of vertices in V that are expressed by w, e.g., the WordNet synsets that include w as one of their senses.

Train-O-Matic consists of three steps:

- **Lexical profiling:** for each vertex in the semantic network, we compute its Personalized PageRank vector, which provides its lexical-semantic profile (Section 2.1).
- **Sentence scoring:** For each sentence containing a word w, we compute a probability distribution over all the senses of w based on its context (Section 2.2).
- **Sentence ranking and selection:** for each sense s of a word w in the vocabulary, we select those sentences that are most likely to use w in the sense of s (Section 2.3).

2.1 Lexical profiling

In terms of semantic networks the probability of reaching a node v' starting from v can be interpreted as a measure of relatedness between the synsets v and v'. Thus we define the lexical profile of a vertex v in a graph $G = (V, E)$ as the probability distribution over all the vertices v' in the graph. Such distribution is computed by applying the Personalized PagaRank algorithm, a variant of the traditional PageRank (Brin and Page, 1998). While the latter is equivalent to performing random walks with uniform restart probability on every vertex at each step, PPR, on the other hand, makes the restart probability non-uniform, thereby concentrating more probability mass in the surroundings of those vertices having higher restart probability. Formally, (P)PR is computed as follows:

$$v^{(t+1)} = (1 - \alpha)v^{(0)} + \alpha M v^{(t)} \quad (1)$$

where M is the row-normalized adjacency matrix of the semantic network, the restart probability distribution is encoded by vector $v^{(0)}$, and α is the well-known damping factor usually set to 0.85 (Brin and Page, 1998). If we set $v^{(0)}$ to a unit probability vector $(0, \ldots, 0, 1, 0, \ldots, 0)$, i.e., restart is always on a given vertex, PPR outputs the probability of reaching every vertex starting from the restart vertex after a certain number of steps. This approach has been used in the literature to create semantic signatures (i.e., profiles) of individual concepts, i.e., vertices of the semantic network (Pilehvar et al., 2013), and then to determine the semantic similarity of concepts. As also done by Pilehvar and Collier (2016), we instead use the PPR vector as an estimate of the conditional probability of a word w' given the target sense[1] $s \in V$ of word w:

$$P(w'|s, w) = \frac{\max_{s' \in Senses(w')} v_s(s')}{Z} \quad (2)$$

where $Z = \sum_{w"} P(w"|s, w)$ is a normalization constant, v_s is the vector resulting from an adequate number of random walks used to calculate PPR, and $v_s(s')$ is the vector component corresponding to sense s'. To fix the number of iterations needed to have a sufficiently accurate vector, we follow Lofgren et al. (2014) and set the error $\delta = 0.00001$ and the number of iterations to $\frac{1}{\delta} = 100,000$.

As a result of this lexical profiling step we have a probability distribution over vocabulary words for each given word sense of interest.

2.2 Sentence scoring

The objective of the second step is to score the importance of word senses for each of the corpus sentences which contain the word of interest. Given a sentence $\sigma = w_1, w_2, \ldots, w_n$, for a given target word w in the sentence ($w \in \sigma$), and for each of its senses $s \in Senses(w)$, we compute the probability $P(s|\sigma, w)$. Thanks to Bayes' theorem we can determine the probability of sense s of w given the

[1] Note that we use senses and concepts (synsets) interchangeably, because – given a word – a word sense unambiguously determines a concept (i.e., the synset it is contained in) and vice versa.

sentence as follows:

$$P(s|\sigma,w) = \frac{P(\sigma|s,w)P(s|w)}{P(\sigma|w)} \quad (3)$$

$$= \frac{P(w_1,\ldots,w_n|s,w)P(s|w)}{P(w_1,\ldots,w_n|w)}$$

$$\propto P(w_1,\ldots,w_n|s,w)P(s|w) \quad (4)$$

$$\approx P(w_1|s,w)\ldots P(w_n|s,w)P(s|w) \quad (5)$$

where Formula 4 is proportional to the original probability (due to removing the constant in the denominator) and is approximated with Formula 5 due to the assumption of independence of the words in the sentence. $P(w_i|s,w)$ is calculated as in Formula 2 and $P(s|w)$ is set to $1/|Senses(w)|$ (recall that s is a sense of w). For example, given the sentence σ = "A match is a tool for starting a fire", the target word w = *match* and its set of senses $S_{match} = \{s^1_{match}, s^2_{match}\}$, where s^1_{match} is the sense of *lighter* and s^2_{match} is the sense of *game match*, we want to calculate the probability of each $s^i_{match} \in S_{match}$ of being the correct sense of *match* in the sentence σ. Following Formula 5 we have:

$$\begin{aligned}
&P(s^1_{match}|\sigma, match) \approx \\
&\quad P(\text{tool}|s^1_{match}, match) \\
&\quad \cdot P(\text{start}|s^1_{match}, match) \\
&\quad \cdot P(\text{fire}|s^1_{match}, match) \\
&\quad \cdot P(s^1_{match}|match) \\
&\quad = 2.1 \cdot 10^{-4} \cdot 2 \cdot 10^{-3} \cdot 10^{-2} \cdot 5 \cdot 10^{-1} \\
&\quad = 2.1 \cdot 10^{-9}
\end{aligned}$$

$$\begin{aligned}
&P(s^2_{match}|\sigma, match) \approx \\
&\quad P(\text{tool}|s^2_{match}, match) \\
&\quad \cdot P(\text{start}|s^2_{match}, match) \\
&\quad \cdot P(\text{fire}|s^2_{match}, match) \\
&\quad \cdot P(s^2_{match}|match) \\
&\quad = 10^{-5} \cdot 2.9 \cdot 10^{-4} \cdot 10^{-6} \cdot 5 \cdot 10^{-1} \\
&\quad = 1.45 \cdot 10^{-15}
\end{aligned}$$

As can be seen, the first sense of *match* has a much higher probability due to its stronger relatedness to the other words in the context (i.e. *start*, *fire* and *tool*). Note also that all the probabilities for the second sense are at least one magnitude less than the probability of the first sense.

2.3 Sense-based sentence ranking and selection

Finally, for a given word w and a given sense $s_1 \in Senses(w)$, we score each sentence σ in which w appears and s_1 is its most likely sense according to a formula that takes into account the difference between the first (i.e., s_1) and the second most likely sense of w in σ:

$$\Delta_{s_1}(\sigma) = P(s_1|\sigma,w) - P(s_2|\sigma,w) \quad (6)$$

where $s_1 = \arg\max_{s \in Senses(w)} P(s|\sigma,w)$, and $s_2 = \arg\max_{s \in Senses(w)\setminus\{s_1\}} P(s|\sigma,w)$. We then sort all sentences based on $\Delta_{s_1}(\cdot)$ and return a ranked list of sentences where word w is most likely to be sense-annotated with s_1. Although we recognize that other scoring strategies could have been used, this was experimentally the most effective one when compared to alternative strategies, i.e., the sense probability, the number of words related to the target word w, the sentence length or a combination thereof.

3 Creating a Denser and Multilingual Semantic Network

In the previous Section we assumed that WordNet was our semantic network, with synsets as vertices and edges represented by its semantic relations. However, while its lexical coverage is high, with a rich set of fine-grained synsets, at the relation level WordNet provides mainly paradigmatic information, i.e., relations like hypernymy (is-a) and meronymy (part-of). It lacks, on the other hand, syntagmatic relations, such as those that connect verb synsets to their arguments (e.g., the appropriate senses of *eat*$_v$ and *food*$_n$), or pairs of noun synsets (e.g., the appropriate senses of *bus*$_n$ and *driver*$_n$).

Intuitively, Train-O-Matic would suffer from such a lack of syntagmatic relations, as the relevance of a sense for a given word in a sentence depends directly on the possibility of visiting senses of the other words in the same sentence (cf. Formula 5) via random walks as calculated with Formula 1. Such reachability depends on the connections available between synsets. Because syntagmatic relations are sparse in WordNet, if it was used on its own, we would end up with a poor ranking of sentences for any given word sense. Moreover, even though the methodology presented in Section 2 is language-independent, Train-O-Matic would lack informa-

mouse (animal)		mouse (device)	
WordNet	WordNet$_{BN}$	WordNet	WordNet$_{BN}$
mouse^1_n	mouse^1_n	mouse^4_n	mouse^4_n
tail^1_n	little^1_a	wheel^1_n	computer^1_n
hairless^1_a	rodent^1_n	electronic_device^1_n	pad^4_n
rodent^1_n	cheese^1_n	ball^3_n	cursor^1_n
trunk^3_n	cat^1_n	hand_operated^1_n	operating_system^1_n
elongate^2_a	rat^1_n	mouse_button^1_n	trackball^1_n
house_mouse^1_n	elephant^1_n	cursor^1_n	wheel^1_n
minuteness^1_n	pet^1_n	operate^3_v	joystick^1_n
nude_mouse^1_n	experiment^1_n	object^1_n	Windows^1_n

Table 1: Top-ranking synsets of the PPR vectors computed on WordNet (first and third columns) and WordNet$_{BN}$ (second and fourth columns) for *mouse* as animal (left) and as device (right).

tion (e.g. senses for a word in an arbitrary vocabulary) for languages other than English.

To cope with these issues, we exploit BabelNet,[2] a huge multilingual semantic network obtained from the automatic integration of WordNet, Wikipedia, Wiktionary and other resources (Navigli and Ponzetto, 2012), and create the BabelNet subgraph induced by the WordNet vertices. The result is a graph whose vertices are BabelNet synsets that contain at least one WordNet synset and whose edge set includes all those relations in BabelNet coming either from WordNet itself or from links in other resources mapped to WordNet (such as hyperlinks in a Wikipedia article connecting it to other articles). The greatest contribution of syntagmatic relations comes, indeed, from Wikipedia, as its articles are linked to related articles (e.g., the English Wikipedia *Bus* article[3] is linked to *Passenger*, *Tourism*, *Bus lane*, *Timetable*, *School*, and many more).

Because not all Wikipedia (and other resources') pages are connected with the same degree of relatedness (e.g., countries are often linked, but they are not necessarily closely related to the source article in which the link occurs), we apply the following weighting strategy to each edge $(s, s') \in E$ of our WordNet-induced subgraph of BabelNet $G = (V, E)$:

$$w(s,s') = \begin{cases} 1 & (s,s') \in E(\text{WordNet}) \\ WO(s,s') & \text{otherwise} \end{cases} \quad (7)$$

where E(WordNet) is the edge set of the original WordNet graph and $WO(s, s')$ is the weighted overlap measure which calculates the similarity between two synsets:

$$WO(s,s') = \frac{\sum_{i=1}^{|S|}(r_i^1 + r_i^2)^{-1}}{\sum_{i=1}^{|S|}(2i)^{-1}}$$

where r_i^1 and r_i^2 are the rankings of the i-th synsets in the set S of the components in common between the vectors associated with s and s', respectively. Because at this stage we still have to calculate our synset vector representation, we use the precomputed NASARI vectors (Camacho-Collados et al., 2015) to calculate WO. This choice is due to WO's higher performance over cosine similarity for vectors with explicit dimensions (Pilehvar et al., 2013).

As a result, each row of the original adjacency matrix M of G will be replaced with the weights calculated in Formula 7 and then normalized in order to be ready for PPR calculation (see Formula 1). An idea of why a denser semantic network has more useful connections and thus leads to better results is provided by the example in Table 1[4], where we show the highest-probability synsets in the PPR vectors calculated with Formula 1 for two different senses of *mouse* (its animal and device senses) when WordNet (left) and our WordNet-induced subgraph of BabelNet (WordNet$_{BN}$, right) are used as the underlying semantic network for PPR computation. Note that WordNet's top synsets are related to the target synset via paradigmatic (i.e., hypernymy and meronymy) relations, while WordNet$_{BN}$ includes many syntagmatically-related synsets (e.g., *exper-*

[2] http://babelnet.org

[3] Retrieved on February 3rd, 2017.

[4] We use the notation w_p^k introduced in (Navigli, 2009) to denote the k-th sense of word w with part-of-speech tag p.

iment for the animal, and *operating system* and *Windows* for the device sense, among others).

4 Experimental Setup

Corpora for sense annotation We used two different corpora to extract sentences: Wikipedia and the United Nations Parallel Corpus (Ziemski et al., 2016). The first is the largest and most up-to-date encyclopedic resource, containing definitional information, the second, on the other hand, is a public collection of parliamentary documents of the United Nations. The application of Train-O-Matic to the two corpora produced two sense-annotated datasets, which we named T-O-M$_{Wiki}$ and T-O-M$_{UN}$, respectively.

Semantic Network We created sense-annotated corpora with Train-O-Matic both when using PPR vectors computed from vanilla WordNet and when using WordNet$_{BN}$, our denser network obtained from the WordNet-induced subgraph of BabelNet (see Section 3).

Gold standard datasets We performed our evaluations using the framework made available by Raganato et al. (2017a) on five different all-words datasets, namely: the Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), SemEval-2007 (Pradhan et al., 2007), SemEval-2013 (Navigli et al., 2013) and SemEval-2015 (Moro and Navigli, 2015) WSD datasets. We focused on nouns only, given the fact that Wikipedia provides connections between nominal synsets only, and therefore contributes mainly to syntagmatic relations between nouns.

Comparison sense-annotated corpora To show the impact of our T-O-M corpora in WSD, we compared its performance on the above gold standard datasets, against training with:

- **SemCor** (Miller et al., 1993), a corpus containing about 226,000 words annotated manually with WordNet senses.
- **One Million Sense-Tagged Instances** (Taghipour and Ng, 2015, OMSTI), a sense-annotated dataset obtained via a semi-automatic approach based on the disambiguation of a parallel corpus, i.e., the United Nations Parallel Corpus, performed by exploiting manually translated word senses. Because OMSTI integrates SemCor to increase coverage, to keep a level playing field we excluded the latter from the corpus.

We note that T-O-M, instead, is fully automatic and does not require any WSD-specific human intervention nor any aligned corpus.

Reference system In all our experiments, we used It Makes Sense (Zhong and Ng, 2010, IMS), a state-of-the-art WSD system based on linear Support Vector Machines, as our reference system for comparing its performance when trained on T-O-M, against the same WSD system trained on other sense-annotated corpora (i.e., SemCor and OMSTI). Following the WSD literature, unless stated otherwise, we report performance in terms of F1, i.e., the harmonic mean of precision and recall.

We note that it is not the purpose of this paper to show that T-O-M, when integrated into IMS, beats all other configurations or alternative systems, but rather to fully automatize the WSD pipeline with performances which are competitive with the state of the art.

Baseline As a traditional baseline in WSD, we used the Most Frequent Sense (MFS) baseline given by the first sense in WordNet. The MFS is a very competitive baseline, due to the sense skewness phenomenon in language (Navigli, 2009).

Number of training sentences per sense Given a target word w, we sorted its senses $Senses(w)$ following the WordNet ordering and selected the top k_i training sentences for the i-th sense according to Formula 6, where:

$$k_i = \frac{1}{i^z} * K \tag{8}$$

with $K = 500$ and $z = 2$ which were tuned on a separate small in-house development dataset[5].

5 Results

5.1 Impact of syntagmatic relations

The first result we report regards the impact of vanilla WordNet vs. our WordNet-induced subgraph of BabelNet (WordNet$_{BN}$) when calculating PPR vectors. As can be seen from Table 2 – which shows the performance of the T-O-M$_{Wiki}$ corpora generated with the two semantic networks – using WordNet for PPR computation decreases

[5] 50 word-sense pairs annotated manually.

Dataset	T-O-M$_{Wiki}$ BN	T-O-M$_{Wiki}$ WN
Senseval-2	**70.5**	70.0
Senseval-3	**67.4**	63.1
SemEval-07	**59.8**	57.9
SemEval-13	**65.5**	63.7
SemEval-15	68.6	**69.5**
ALL	**67.3**	65.7

Table 2: F1 of IMS trained on T-O-M when PPR is obtained from the WordNet graph (WN) and from the WordNet-induced subgraph of BabelNet (BN).

the overall performance of IMS from 0.5 to around 4 points across the five datasets, with an overall loss of 1.6 F1 points. Similar performance losses were observed when using T-O-M$_{UN}$ (see Table 3). This corroborates our hunch discussed in Section 3 that a resource like BabelNet can contribute important syntagmatic relations that are beneficial for identifying (and ranking high) sentences which are semantically relevant for the target word sense. In the following experiments, we report only results using WordNet$_{BN}$.

5.2 Comparison against sense-annotated corpora

We now move to comparing the performance of T-O-M, which is fully automatic, against corpora which are annotated manually (SemCor) and semi-automatically (OMSTI). In Table 3 we show the F1-score of IMS on each gold standard dataset in the evaluation framework and on all datasets merged together (last row), when it is trained with the various corpora described above.

As can be seen, T-O-M$_{Wiki}$ and T-O-M$_{UN}$ obtain higher performance than OMSTI (up to 5.5 points above) on 3 out of 5 datasets, and, overall, T-O-M$_{Wiki}$ scores 1 point above OMSTI. The MFS is in the same ballpark as T-O-M$_{Wiki}$, performing better on some datasets and worse on others. We note that IMS trained on T-O-M$_{Wiki}$ succeeds in surpassing or obtaining the same results as IMS trained on SemCor on SemEval-15 and SemEval-13. We view this as a significant achievement given the total absence of manual effort involved in T-O-M. Because overall T-O-M$_{Wiki}$ outperforms T-O-M$_{UN}$, in what follows we report all the results with T-O-M$_{Wiki}$, except for the domain-oriented evaluation (see Section 5.4).

5.3 Performance without backoff strategy

IMS uses the MFS as a backoff strategy when no sense can be output for a target word in context (Zhong and Ng, 2010). Consequently, the performance of the MFS is mixed up with that of the SVM classifier. As shown in Table 4, OMSTI is able to provide annotated sentences for roughly half of the tokens in the datasets. Train-O-Matic, on the other hand, is able to cover almost all words in each dataset with at least one training sentence. This means that in around 50% of cases OMSTI gives an answer based on the IMS backoff strategy.

To determine the real impact of the different training data, we therefore decided to perform an additional analysis of the IMS performance when the MFS backoff strategy is disabled. Because we suspected the system would not always return a sense for each target word, in this experiment we measured precision, recall and their harmonic mean, i.e., F1. The results in Table 5 confirm our hunch, showing that OMSTI's recall drops heavily, thereby affecting F1 considerably. T-O-M performances, instead, remain high in terms of precision, recall and F1. This confirms that OMSTI relies heavily on data (those obtained for the MFS and from SemCor) that are produced manually, rather than semi-automatically.

5.4 Domain-oriented WSD

To further inspect the ability of T-O-M to enable disambiguation in different domains, we decided to evaluate on specific documents from the various gold standard datasets which could be clearly assigned a domain label. Specifically, we tested on 13 SemEval-13 documents from various domains[6] and 2 SemEval-15 documents (namely, maths & computers, and biomedicine) and carried out two separate tests and evaluations of T-O-M on each domain: once using the MFS backoff strategy, and once not using it. In Tables 6 and 7 we report the results of both T-O-M$_{Wiki}$ and T-O-M$_{UN}$ to determine the impact of the corpus type.

As can be seen in the tables, T-O-M$_{Wiki}$ systematically attains higher scores than OMSTI (except for the biology domain), and, in most cases, attains higher scores than MFS when the backoff is used, with a drastic, systematic increase over OMSTI with both Train-O-Matic configurations

[6]Namely biology, climate, finance, health care, politics, social issues and sport.

Dataset	Train-O-Matic$_{Wiki}$	Train-O-Matic$_{UN}$	OMSTI	SemCor	MFS
Senseval-2	70.5	69.0	74.1	**76.8**	72.1
Senseval-3	67.4	68.3	67.2	**73.8**	72.0
SemEval-07	59.8	57.9	62.3	**67.3**	65.4
SemEval-13	**65.5**	62.5	62.8	**65.5**	63.0
SemEval-15	**68.6**	63.5	63.1	66.1	66.3
ALL	67.3	65.3	66.4	**70.4**	67.6

Table 3: F1 of IMS trained on Train-O-Matic, OMSTI and SemCor, and MFS for the Senseval-2, Senseval-3, SemEval-07, SemEval-13 and SemEval-15 datasets.

Dataset	OMSTI	Train-O-Matic	Total
Senseval-2	469	1005	1066
Senseval-3	494	860	900
Semeval-07	89	159	159
Semeval-13	757	1428	1644
Semeval-15	249	494	531
ALL	2058	3946	4300

Table 4: Number of nominal tokens for which at least one training example is provided by OMSTI or Train-O-Matic for each dataset.

Dataset	OMSTI			Train-O-Matic		
	P	R	F1	P	R	F1
Senseval-2	64.8	28.5	39.6	69.5	65.5	**67.4**
Senseval-3	55.7	31.0	39.8	66.1	63.1	**64.6**
SemEval-07	64.1	35.9	46.0	59.8	59.8	**59.8**
SemEval-13	50.7	23.4	32.0	61.3	53.3	**57.0**
SemEval-15	57.0	26.7	36.4	67.0	62.3	**64.6**
ALL	56.5	27.0	36.5	65.1	59.7	**62.3**

Table 5: Precision, Recall and F1 of IMS trained on OMSTI and Train-O-Matic corpus without MFS backoff strategy for Senseval-2, Senseval-3, SemEval-07, SemEval-13 and SemEval-15.

in recall and F1 when the backoff strategy is disabled. This demonstrates the usefulness of the corpora annotated by Train-O-Matic not only on open text, but also on specific domains. We note that T-O-M$_{UN}$ obtains the best results in the politics domain, which is the closest domain to the UN corpus from which its training sentences are obtained.

6 Scaling up to Multiple Languages

Experimental Setup In this section we investigate the ability of Train-O-Matic to scale to low-resourced languages, such as Italian and Spanish, for which training data for WSD is not available.

Thanks to BabelNet, in fact, Train-O-Matic can be used to generate sense-annotated data for any language supported by the knowledge base. Thus, in order to build new training datasets for the two languages, we ran Train-O-Matic on their corresponding versions of Wikipedia, then we tuned the two parameters K and z on an in-house development dataset[7]. In contrast to the English setting, in order to calculate Formula 8 we sorted the senses of each word by vertex degree. Finally we used the output data to train IMS.

Results To perform our evaluation we chose the most recent multilingual task (SemEval 2015 task 13) which includes gold data for Italian and Spanish. As can be seen from Table 8 Train-O-Matic enabled IMS to perform better than the best participating system (Manion and Sainudiin, 2014, SUDOKU) in all three settings (All domains, Maths & Computer and Biomedicine). Its performance was in fact, 1 to 3 points higher, with a 6-point peak on Maths & Computer in Spanish and on Biomedicine in Italian. This demonstrates the ability of Train-O-Matic to enable supervised WSD systems to surpass state-of-the-art knowledge-based WSD approaches in low-resourced languages without relying on manually curated data for training.

7 Related Work

There are two mainstream approaches to Word Sense Disambiguation: supervised and knowledge-based approaches. Both suffer in different ways from the so-called knowledge acquisition bottleneck, that is, the difficulty in obtaining an adequate amount of lexical-semantic data: for training in the case of supervised systems, and for enriching semantic networks in the case of knowledge-based ones (Pilehvar and

[7]We set $K = 100$ and $z = 2.3$ for Spanish and $K = 100$ and $z = 2.5$ for Italian.

Domain	Backoff	T-O-M$_{Wiki}$			T-O-M$_{UN}$			OMSTI			SemCor	MFS	Size
		P	R	F1	P	R	F1	P	R	F1	F1	F1	
Biology	MFS	63.0	63.0	63.0	65.9	65.9	**65.9**	65.9	65.9	**65.9**	66.3	64.4	135
	-	59.0	53.3	56.0	62.3	56.3	**59.2**	48.1	18.5	26.7	-		
Climate	MFS	68.1	68.1	**68.1**	63.4	63.4	63.4	68.0	68.0	68.0	70.1	67.5	194
	-	63.4	50.0	**55.9**	57.5	45.4	50.7	58.0	24.2	34.2	-		
Finance	MFS	68.0	68.0	**68.0**	56.6	56.6	56.6	64.4	64.4	64.4	63.7	56.2	219
	-	62.1	51.6	**56.4**	48.4	40.2	43.9	57.4	28.3	37.9	-		
Health Care	MFS	65.2	65.2	**65.2**	60.1	60.1	60.1	52.9	52.9	52.9	62.7	56.5	138
	-	61.3	55.1	**58.0**	55.6	50.0	52.6	34.6	18.4	24.0	-		
Politics	MFS	65.2	65.2	65.2	66.3	66.3	**66.3**	63.4	63.4	63.4	69.5	67.7	279
	-	62.5	54.8	58.4	63.9	55.9	**59.6**	54.1	21.5	30.8	-		
Social Issues	MFS	68.5	68.5	**68.5**	63.6	63.6	63.6	65.6	65.6	65.6	66.8	67.6	349
	-	63.1	53.0	**57.6**	57.2	47.9	52.1	54.7	25.2	34.5	-		
Sport	MFS	60.3	60.3	60.3	60.9	60.9	**60.9**	58.8	58.8	58.8	60.4	57.6	330
	-	58.3	54.6	**56.4**	58.1	53.3	55.5	45.0	23.0	30.4	-		

Table 6: Performance comparison over SemEval-2013 domain-specific datasets.

Domain	Backoff	T-O-M$_{Wiki}$			T-O-M$_{UN}$			OMSTI			SemCor	MFS	Size
		P	R	F1	P	R	F1	P	R	F1	F1	F1	
Biomedicine	MFS	76.3	76.3	**76.3**	66.0	66.0	66.0	64.9	64.9	64.9	70.3	71.1	100
	-	76.1	72.2	**74.1**	64.4	59.8	62.0	60.5	26.8	37.2	-		
Maths & Computer	MFS	50.0	50.0	**50.0**	48.0	48.0	48.0	36.0	36.0	36.0	40.6	40.9	97
	-	50.0	47.0	**48.5**	47.8	44.0	45.8	21.2	11.0	14.5	-		

Table 7: Performance comparison over the Biomedical and Maths & Computer domains in SemEval-15.

Language	Dataset	Best System	Train-O-Matic		
		F1	P	R	F1
Italian	ALL	56.6	65.1	55.6	**59.9**
	Computers & Math	46.6	52.7	43.3	**47.6**
	Biomedicine	65.9	76.6	67.6	**71.8**
Spanish	ALL	56.3	61.3	54.8	**57.9**
	Computers & Math	42.4	53.3	44.4	**48.5**
	Biomedicine	65.5	71.8	65.5	**68.5**

Table 8: Performance comparison between T-O-M and SemEval-2015's best SUDOKU Run.

Navigli, 2014; Navigli, 2009).

State-of-the-art supervised systems include Support Vector Machines such as IMS (Zhong and Ng, 2010) and, more recently, LSTM neural networks with attention and multitask learning (Raganato et al., 2017b) as well as LSTMs paired with nearest neighbours classification (Melamud et al., 2016; Yuan et al., 2016). The latter also integrates a label propagation algorithm in order to enrich the sense annotated dataset. The main difference from our approach is its need for a manually annotated dataset to start the label propagation algorithm, whereas Train-O-Matic is fully automatic. An evaluation against this system would have been interesting, but neither the proprietary training data nor the code are available at the time of writing.

In order to generalize effectively, these supervised systems require large numbers of training instances annotated with senses for each target word occurrence. Overall, this amounts to millions of training instances for each language of interest, a number that is not within reach for any language. In fact, no supervised system has been submitted in major multilingual WSD competitions for languages other than English (Navigli et al., 2013; Moro and Navigli, 2015). To overcome this problem, new methodologies have recently been developed which aim to create sense-tagged corpora automatically. Raganato et al. (2016) developed 7 heuristics to grow the number of hyperlinks in Wikipedia pages. Otegi et al. (2016) applied a different disambiguation pipeline for each language to parallel text in Europarl (Koehn, 2005) and QTLeap (Agirre et al., 2015) in order to enrich them with semantic annotations. Taghipour and Ng (2015), the work closest to ours, exploits the alignment from English to Chinese sentences of

the United Nation Parallel Corpus (Ziemski et al., 2016) to reduce the ambiguity of English words and sense-tag English sentences. The assumption is that the second language is less ambiguous than the first one and that hand-made translations of senses are available for each WordNet synset. This approach is, therefore, semi-automatic and relies on certain assumptions, in contrast to Train-O-Matic which is, instead, fully automatic and can be applied to any kind of corpus (and language) depending on the specific need. Earlier attempts at the automatic extraction of training samples were made by Agirre and De Lacalle (2004) and Fernández et al. (2004). Both exploited the monosemous relatives method (Leacock et al., 1998) in order to retrieve sentences from the Web which contained a given monosemous noun or a relative monosemous word (e.g., a synonym, a hypernym, etc.). As can be seen in (Fernández et al., 2004) this approach can lead to the retrieval of very accurate examples, but its main drawback lies in the number of senses covered. In fact, for all those synsets that do not have any monosemous relative, the system is unable to retrieve examples, thus heavily affecting the performance in terms of recall and F1.

Knowledge-based WSD, instead, bypasses the heavy requirement of sense-annotated corpora by applying algorithms that exploit a general-purpose semantic network, such as WordNet, which encodes the relational information that interconnects synsets via different kinds of relation. Approaches include variants of Personalized PageRank (Agirre et al., 2014) and densest subgraph approximation algorithms (Moro et al., 2014) which, thanks to the availability of multilingual resources such as BabelNet, can easily be extended to perform WSD in arbitrary languages. Other approaches to knowledge-based WSD exploit the definitional knowledge contained in a dictionary. The Lesk algorithm (Lesk, 1986) and its variants (Banerjee and Pedersen, 2002; Kilgarriff and Rosenzweig, 2000; Vasilescu et al., 2004) aim to determine the correct sense of a word by comparing each word-sense definition with the context in which the target word appears. The limit of knowledge-based WSD, however, lies in the absence of mechanisms that can take into account the very local context of a target word occurrence, including non-content words such as prepositions and articles. Furthermore, recent studies seem to suggest that such approaches are barely able to surpass supervised WSD systems when they enrich their networks starting from a comparable amount of annotated data (Pilehvar and Navigli, 2014). With T-O-M, rather than further enriching an existing semantic network, we exploit the information available in the network to annotate raw sentences with sense information and train a state-of-the-art supervised WSD system without task-specific human annotations.

8 Conclusion

In this paper we presented Train-O-Matic, a novel approach to the automatic construction of large training sets for supervised WSD in an arbitrary language. Train-O-Matic removes the burden of manual intervention by leveraging the structural semantic information available in the WordNet graph enriched with additional relational information from BabelNet, and achieves performance competitive to that of semi-automatic approaches and, in some cases, of manually-curated training data. T-O-M was shown to provide training data for virtually all the target ambiguous nouns, in marked contrast to alternatives like OMSTI, which covers in many cases around half of the tokens, resorting to the MFS otherwise. Moreover Train-O-Matic has proven to scale well to low-resourced languages, for which no manually annotated dataset exists, surpassing the current state of the art of knowledge-based systems.

We believe that the ability of T-O-M to overcome the current paucity of annotated data for WSD, coupled with video games with a purpose for validation purposes (Jurgens and Navigli, 2014; Vannella et al., 2014), paves the way for high-quality multilingual supervised WSD. All the training corpora, including approximately one million sentences which cover English, Italian and Spanish, are made available to the community at `http://trainomatic.org`.

As future work we plan to extend our approach to verbs, adjectives and adverbs. Following Bennett et al. (2016) we will also experiment on more realistic estimates of $P(s|w)$ in Formula 5 as well as other assumptions made in our work.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487.

References

Eneko Agirre, António Branco, Martin Popel, and Kiril Simov. 2015. Europarl QTLeap WSD/NED corpus. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University in Prague.

Eneko Agirre and Oier Lopez De Lacalle. 2004. Publicly available topic signatures for all wordnet nominal senses. In *LREC*.

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

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

Andrew Bennett, Timothy Baldwin, Jey Han Lau, Diana McCarthy, and Francis Bond. 2016. Lexsemtm: A semantic dataset based on all-words unsupervised sense distribution learning. In *Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1513 – 1524, Berlin, Germany.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Nasari: a novel approach to a semantically-aware representation of items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 567–577, Denver, Colorado. Association for Computational Linguistics.

Philip Edmonds and Scott Cotton. 2001. Senseval-2: overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5. Association for Computational Linguistics.

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

Juan Fernández, Mauro Castillo Valdés, German Rigau Claramunt, Jordi Atserias Batalla, and Jordi Tormo. 2004. Automatic acquisition of sense examples using exretriever. In *IBERAMIA Workshop on Lexical Resources and The Web for Word Sense Disambiguation*, pages 97–104.

David Jurgens and Roberto Navigli. 2014. It's All Fun and Games until Someone Annotates: Video Games with a Purpose for Linguistic Annotation. *Transactions of the Association for Computational Linguistics (TACL)*, 2:449–464.

Adam Kilgarriff and Joseph Rosenzweig. 2000. Framework and results for english SENSEVAL. *Computers and the Humanities*, 34(1–2):15–48.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Claudia Leacock, George A Miller, and Martin Chodorow. 1998. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

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.

Peter A Lofgren, Siddhartha Banerjee, Ashish Goel, and C Seshadhri. 2014. Fast-ppr: Scaling personalized pagerank estimation for large graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1445. ACM.

Steve L Manion and Raazesh Sainudiin. 2014. An iterative "sudoku style" approach to subgraph-based word sense disambiguation. In *In Proceedings of the Third Joint Conference on Lexical and Computational Se- mantics (*SEM 2014)*, pages 40–50, Dublin, Ireland.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of CONLL*, pages 51–61.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, N.J.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proc. 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. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of the 7^{th} International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM 2013)*, pages 222–231, Atlanta, USA.

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.

Arantxa Otegi, Nora Aranberri, Antonio Branco, Jan Hajic, Steven Neale, Petya Osenova, Rita Pereira, Martin Popel, Joao Silva, Kiril Simov, et al. 2016. Qtleap wsd/ned corpora: Semantic annotation of parallel corpora in six languages. In *Proceedings of the 10th Language Resources and Evaluation Conference, LREC*, pages 3023–3030.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, TX.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of ACL*, pages 1341–1351.

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):837–881.

Sameer S Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92.

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

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.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of Empirical Methods in Natural Language Processing*, Copenhagen, Denmark.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain.

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

Daniele Vannella, David Jurgens, Daniele Scarfini, Domenico Toscani, and Roberto Navigli. 2014. Validating and extending semantic knowledge bases using video games with a purpose. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 1294–1304, Baltimore, Maryland. Association for Computational Linguistics.

Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. 2004. Evaluating variants of the lesk approach for disambiguating words. In *Proceedings of LREC*, Lisbon, Portugal.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. *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 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.

Micha Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portoroz, Slovenia. European Language Resources Association (ELRA).