UFSAC: Unification of Sense Annotated Corpora and Tools

Loïc Vial, Benjamin Lecouteux, Didier Schwab
LIG – GETALP – Univ. Grenoble Alpes – France
{loic.vial, benjamin.lecouteux, didier.schwab}@imag.fr

Abstract

In Word Sense Disambiguation, sense annotated corpora are often essential for evaluating a system and also valuable in order to reach a good efficiency. Always created for a specific purpose, there are today a dozen of sense annotated English corpora, in various formats and using different versions of WordNet. The main hypothesis of this work is that it should be possible to build a disambiguation system by using any of these corpora during the training phase or during the testing phase regardless of their original purpose. In this article, we present UFSAC: a format of corpus that can be used for either training or testing a disambiguation system, and the process we followed for constructing this format. We give to the community the whole set of sense annotated English corpora that we know, in this unified format, when the copyright allows it, with sense keys converted to the last version of WordNet. We also provide the source code for building these corpora from their original data, and a complete Java API for manipulating corpora in this format. The whole resource is available at the following URL: `https://github.com/getalp/UFSAC`.

Keywords: Word Sense Disambiguation, sense annotated corpora, unified resource, tools

1. Introduction

Whether they are used for the evaluation or for the learning process of a Word Sense Disambiguation (WSD) system, the importance of sense annotated corpora in Natural Language Processing (NLP) is considerable. On one hand, the evaluation *in vivo*, i.e. the evaluation of a WSD system as part of a larger task, has never been really exploited. On the other hand, the evaluation *in vitro*, which uses directly sense annotated corpora by comparing the output of a system to manual annotations, is predominant. Moreover, WSD systems exploiting examples from sense annotated corpora are generally far better than those which do not (Navigli et al., 2007; Moro and Navigli, 2015).

At the time of its creation, WordNet (Miller, 1995) was undoubtedly the only lexical database freely available for English. Since the beginning of the 2000s, it has become the *de facto* standard for WSD in this language. Indeed, most of sense annotated corpora are either directly annotated with WordNet sense keys or they are annotated with a sense inventory linked to the senses of WordNet, such as BabelNet (Navigli and Ponzetto, 2010).

However, it is not trivial to use these corpora, because most of them differ in their format and on the version of WordNet they use. As a consequence, very few works in the literature of WSD are trained or evaluated on more than two annotated corpora.

Also, WSD systems are systematically evaluated on corpora that have been initially created for the purpose of evaluation, and never on corpora that have been created for another purpose, such as training or for sense distribution estimation, whereas there is no scientific reason for that.

This paper presents a work of unification of all existing English corpora annotated with any version of WordNet to our knowledge, in a unique format, easy to understand, and easy to work with in practice. We put on the same level the corpora originally created for the evaluation and those for the learning, so to facilitate the creation of robust WSD systems which could for example be evaluated in a way where all corpora except one are used for the learning, and the remaining one is used for the evaluation, then switch the corpora and do this for every existing corpus.

The language resource that we provide contains all English sense annotated corpora in UFSAC (Unified Format for Sense Annotated Corpora), the format that we propose, with sense annotations converted to the last version of WordNet (3.0), along with Java code to easily read, write and modify any corpus in this format, and scripts for converting a corpus from its original format to UFSAC.

Our work is the continuity of the demonstration of (Vial et al., 2017), and it differs from the recent work of (Raganato et al., 2017) in several points. Their work is focused on the evaluation of WSD systems, whereas we provide a complete API for manipulating corpora in a new unified format (UFSAC), and conversion scripts allowing the full reconstruction of the corpora from the original data. We also propose five additional corpora in our resource among the most difficult to parse.

In our resource, we provide a script for converting a corpus from our format to theirs, so existing WSD systems that rely on their format can be trained or evaluated on any of the corpus that we produced. We also provide a script for converting their format to ours in order to facilitate any collaborative work in the community.

2. Sense Annotated Corpora: rare and costly resources

Generally speaking, a corpus is a collection of documents which can be used as samples of text for a particular language (Habert et al., 1998). A corpus may contain several millions of words, which can be lemmatized and annotated with information concerning their part of speech for example. Among these corpora, we can find the *British National Corpus* (Burnard, 1998) (100 million words) and the *American National Corpus* (Ide and Macleod, 2001) (20 million words). The texts come from several sources such as newspapers, books, encyclopedias or from the Web.

A sense annotated corpus is a corpus in which some or all words are annotated with an identifier of sense from

a specific lexical database. For example, all words in the corpus of the 7th task of the SemEval 2007 semantic evaluation campaign (Navigli et al., 2007) are annotated with sense identifiers from WordNet 2.1, whereas in the English corpus of the 13th task of SemEval 2015 (Moro and Navigli, 2015), all words are annotated with sense identifiers from WordNet 3.0, BabelNet 2.5 and Wikipedia pages.

There are at least three reasons to create a sense annotated corpus:

- Estimate the distribution of senses in the language. It is for this purpose that the SemCor (Miller et al., 1993) was annotated. Consequently, the senses in WordNet are, since version 1.7, sorted by this distribution of senses estimated on the SemCor.
- Build a Word Sense Disambiguation system which learns from examples contained in the annotated corpus. For instance, the OMSTI (Taghipour and Ng, 2015) was created for this purpose.
- Evaluate a WSD system by comparing its output to the annotations in the corpus, as it is the case for instance with corpora created as part of the evaluation campaigns SensEval-SemEval.

After their distribution, there is no scientific reason not to use indistinctly these corpora either for building a WSD system, for estimating the distribution of senses or for evaluating a WSD system. Indeed, the SemCor is used since a long time for the learning of WSD systems (Chan et al., 2007; Navigli et al., 2007) or more recently for the evaluation of different methods (Yuan et al., 2016). This last usage is still very rare, since it is one of the first experiment that we found in the literature, along with (Màrquez et al., 2002).

However, the format of the resources differs greatly depending on their original purpose. For the SemCor, a single file groups all the information, whereas in the case of the evaluation corpora, there are two files: one that contains the unannotated corpus, and the other that contains the sense annotations. In some corpora, like in the DSO and the OMSTI, there is one file for every lemma in the dictionary, and each file contains thousands of example sentences, where this lemma is the only word that is sense annotated.

Few data are manually sense annotated. The *Global WordNet Association* made a list of 26 corpora annotated with WordNet [1]. These corpora concern 17 languages, but only three of them reach 100,000 annotations. English, with more than 2 million words sense annotated ranks first, before Dutch with nearly 300,000 annotations and Bulgarian with 100,000 annotations. Thus, it is unsurprising that most of researches in WSD focus on English.

3. A single format for sense annotated corpora

The main purpose of this work is to help the construction and the evaluation of WSD systems, by giving to the community the set of all existing English sense annotated corpora to our knowledge, in the same format, using the same sense inventory, and tools to easily parse them, manipulate them, and convert corpora from their original format to our one.

Indeed, a large quantity of sense annotated data is vital for the construction of WSD systems. In evaluation campaigns, this often makes the difference. For example, looking at the data from the SemEval 2007 campaign (Navigli et al., 2007), which most of the recent systems were evaluated on, we observe that systems that did not use sense annotated data obtain a precision score up to $78 - 79\%$[2] (Schwab et al., 2013) (Chen et al., 2014) whereas those which use a lot of annotated data reach a score up to 82% (Chan et al., 2007) (Navigli, 2012) (Vial et al., 2016) and even 84% (Yuan et al., 2016).

Therefore, having all existing corpora in a unique format and using the same sense inventory offers several advantages: it allows to easily expand the quantity of data available for improving WSD systems, it allows to better estimate the distribution of senses in English, and finally, this format can help creating more robust WSD systems. Indeed, we still find a lot of works that focus on a single evaluation task (Vial et al., 2016; Chen et al., 2014), and in these cases, the analysis of the results concerning the robustness of the methods is limited. The unification of the format of sense annotated corpora could improve the evaluation process by facilitating a cross validation process for instance, where the system is evaluated sequentially on every corpus, with all others used for the training.

4. Provided resource

Our work consists in gathering all English corpora sense annotated with WordNet, and convert all of them to a unified format that is able to contain all the informations present in the original format. We created format conversion scripts for this purpose, as well as scripts for cleaning the corpora, and converting the sense annotation to the last version of WordNet (3.0). The resulting corpora are parts of the resource when the copyright allows it, along with the format conversion scripts, the cleaning scripts, and the sense conversion scripts. For the corpora that we cannot distribute because of the licence, anyone that possess them can still run our scripts to turn the original resource into our format. Finally, an API is provided for parsing, creating and manipulating corpora in our format. The resource is accessible at the following URL: `https://github.com/getalp/UFSAC`.

4.1. Sense annotated corpora

Our resource contains the following corpora:

- The *SemCor* (Miller et al., 1993), a subset of the Brown Corpus (Francis and Kučera, 1964). Original annotations are done with WordNet 1.6.
- The *DSO (Defence Science Organisation)* (Ng and Lee, 1997), a non-free corpus, that is focused on 121

[1] `http://globalwordnet.org/wordnet-annotated-corpora/`

[2] This means that the system has chosen the same sense than the human annotators in 78 to 79% of cases

nouns and 70 verbs among the most frequently used and the most ambiguous words in English and have been annotated in various contexts with WordNet 1.5.

- The *WordNet Gloss Tag* [3], a corpus which consists of all definitions of WordNet (Miller, 1995) with every words sense annotated since version 3.0.

- The *OMSTI (One Million Sense-Tagged Instances)* (Taghipour and Ng, 2015), a corpus of approximately one million words sense annotated with WordNet 3.0.

- The *MASC (Manually Annotated Sub-Corpus)* (Ide et al., 2008), we used the version given in the article of (Yuan et al., 2016), annotated with the NOAD (New Oxford American Dictionary), but with corresponding WordNet 3.0 sense keys.

- The *Ontonotes 5.0* (Hovy et al., 2006), annotated with WordNet 3.0.

- The corpora of the WSD evaluation campaigns SemEval-SensEval: SensEval 2 (using WordNet 1.7), SensEval 3 (WN 1.7.1), SemEval 2007 (WN 2.1), SemEval 2013 (WN 3.0) and SemEval 2015 (WN 3.0).

Table 1 summarizes statistics concerning these corpora.
After the conversion of all these corpora into our format, we executed four post-processing steps: sense annotation conversion, identical sentences merging, lemma and POS tagging, and finally a cleaning step.

4.1.1. Sense Annotation Conversion

Sense annotations have been converted, when necessary, from their original WordNet sense key to the last version of WordNet (3.0) thanks to conversion tables from (Daudé et al., 2000).
However, because some senses have been dropped from the old versions of WordNet, some sense annotations have not been converted. In any case, the original sense annotations are always kept alongside the converted sense annotation.
When a sense is mapped to two or more senses with equal probability, all resulting senses are added to the word annotations, separated by a semicolon.

4.1.2. Identical sentences merging

This step is only applied on the DSO and the OMSTI: because these corpora are constructed such that they contain lists of sentences with only one word that is sense annotated, surrounded by words not annotated, some sentences are present in different places across the corpus, but with different words that are sense annotated.
The merging phase identifies identical sentences with annotations on different words, and creates a single sentence containing all annotations. Thus, this steps adds a crucial information for some WSD systems. For instance, a similarity-based WSD system can now "learn" that two word senses are often located in the same sentence.

[3] `http://wordnet.princeton.edu/glosstag.shtml`

4.1.3. Lemma and POS tagging

For the corpora that do not already contain these informations, we added the lemma for every word, when existing, using the WordNet's *morphy* tool, and the part-of-speech tag from the Penn Treebank tag set using Stanford's Log-linear POS tagger (Toutanova et al., 2003).

4.1.4. Cleaning

Finally, this last step consists of trimming words, removing invisible characters and removing inconsistent annotations, for instance when the part of speech annotation differs from the part of speech of the sense annotation.

4.2. UFSAC File format

Our approach for the unification of the different annotated corpora begins with a file format that is descriptive, easily understandable and readable by a human, and at the same time, efficient for a program to parse and create. Finally, it should be able to contain all the information contained in the original resources. These informations are represented with the following concepts:
– A Lexical Entity (LE) is an entity that contains a set of annotations.
– A Corpus is a LE which contains a set of documents.
– A Document is a LE hich contains a set of paragraphs.
– A Paragraph is a LE which contains a set of sentences.
– A Sentence is a LE which contains a set of words.
– A Word is a LE which has a special mandatory annotation "surface form", which is the value of the word.

In order to represent these concepts, UFSAC is based on a simple XML syntax with some conventions: lexical entities are represented by XML nodes (`corpus`, `document`, `paragraph`, `sentence` and `word`), and annotations are node attributes.

The annotations also follow a certain convention, we used the following to annotate words:
– The identifier (`id`) of a lexical entity, particularly useful for corpora originally created for the evaluation (e.g. "d001.s002.t003").
– The surface form (`surface_form`) of a word.
– The lemma (`lemma`) of a word.
– The part of speech (`pos`) of a word.
– The sense of a word, in a specific lexical database, for example WordNet 3.0 (`wn30_key`), WordNet 1.7.1 (`wn171_key`)... If multiple senses are specified (it is the case in the coarse-grained task of SemEval 2007 for instance), they are separated with a semicolon (`;`).
The information of the sense is the one which is the most useful in our case, and it is specific to each lexical database, instead of having a unique "sense" annotation as we can find in most other formats. That way we allow multiples sense annotations from different lexical databases at the same time. For example, the DSO is originally annotated with senses from WordNet 1.5, and the conversion to WordNet 3.0 is sometimes impossible for some senses which were deleted between the two versions. This convention allows us to keep the original annotations, yet to have the annotations from the last version of WordNet, or any other lexical database (for instance BabelNet) at the same time.

Corpus	Sentences	Words		Annotated parts of speech			
		Total	Annotated	Nouns	Verbs	Adj.	Adv.
SemCor	37176	778587	229517	87581	89037	33751	19148
DSO	178119	5317184	176915	105925	70990	0	0
WordNet GlossTag	117659	1634691	496776	232319	62211	84233	19445
MASC	34217	596333	114950	49263	40325	25016	0
OMSTI	820557	35843024	920794	476944	253644	190206	0
Ontonotes	21938	435340	52263	9220	43042	0	0
SemEval 2007 task 07	245	5637	2261	1108	591	356	206
SemEval 2007 task 17	120	3395	455	159	296	0	0
SemEval 2013 task 12	306	8142	1644	1644	0	0	0
SemEval 2015 task 13	138	2638	1053	554	251	166	82
Senseval 2	238	5589	2301	1061	541	422	277
Senseval 3 task 1	300	5511	1957	886	723	336	12

Table 1: Statistics related to our set of annotated corpora, after the conversion and cleaning phase.

The following is an example of the resulting UFSAC XML:

```
<corpus id="short_example">
  <document id="d001" >
    <paragraph>
      <sentence>
        <word surface_form="A" pos="DT" />
        <word surface_form="precise"
           wn30_key="precise%3:00:00::" />
        <word surface_form="example"
              pos="NN" lemma="example" />
        <word surface_form="." />
      </sentence>
    </paragraph>
  </document>
</corpus>
```

Our format thus allows to integrate the whole corpus in a single file, and it is easily readable, especially comparing to most original formats (c.f. the end of section 2.).

4.3. API and tools

An easy-to-use Java API is also provided to read, write and modify efficiently corpora in our format. It allows two styles of programming: you can either load a full corpus in memory, perform all your calculations and save it entirely in a file; or you can sequentially scan, edit or print a corpus from a file, in a streaming manner. The latter is particularly useful when working with huge files which do not fit into memory. Finally, we offer a set of scripts that perform the conversion of a corpus from its original format to our one, and some pre-processing and analyses scripts.

4.3.1. Core API

The core API is a package containing the base classes for manipulating corpora. For simplicity, the class names match exactly what is described in section 4.2..

The class **`Annotation`** describes an annotation on a lexical entity. Concretely, it is a pair of Strings (name/value) and a pointer to the annotated lexical entity.

The class **`LexicalEntity`** describes something that has zero or more annotations, with public methods for accessing/modifying them.

The class **`Word`** inherits from LexicalEntity, has a special mandatory annotation `surface_form`, which is the value of the word, and a parent sentence.

The class **`Sentence`** inherits from LexicalEntity, contains a list of words and a parent paragraph.

The class **`Paragraph`** inherits from LexicalEntity, contains a list of sentences and a parent document.

The class **`Document`** inherits from LexicalEntity, contains a list of paragraphs and a parent corpus.

Finally, the class **`Corpus`** inherits from LexicalEntity and contains a list of documents.

These few classes, coupled with two functions `Corpus.saveToXML` and `Corpus.loadFromXML` allow to create, save, load and modify any corpus easily.

4.3.2. Streaming API

For some corpora particularly huge, like the OMSTI, we also provide a sub-package `streaming`, which allows to read, write or modify a corpus sequentially, without being fully loaded into memory. This is similar to the Java SAX library (Simple API for XML), events are fired when reading a word, sentence, paragraph, etc., and the user can choose to respond to this event or not.

In practice, we provide a set of classes which cover most use cases.

The class **`StreamingCorpusReader`** allows to respond to the events `readBeginCorpus`, `readBeginDocument`, `readWord`, etc.. This can be useful for printing every word that is sense annotated for example.

The class **`StreamingCorpusModifier`** allows to modify a corpus in-place. This is specially useful for preprocessing, for instance convert every word to lowercase.

The class **`StreamingCorpusWriter`** is used for creating a new corpus, with its methods `writeBeginSentence`, `writeWord` and so on.

4.3.3. Scripts

Finally, we provide a set of examples and useful scripts which use our format and our API. The scripts are Java classes with a `main` method and are not part of any package.

The script **`ConvertOriginalCorpora`** allows to convert all corpora listed in subsection 4.1. from their original format to the UFSAC format. This is specially valuable for non-free corpus like the DSO, that we cannot share directly in our format, but that one can still buy in their original format, and then convert to our format. This script includes all post-processing steps described in subsection 4.1..

The scripts **`ConvertFromRaganato`** and **`ConvertToRaganato`** allow to convert a corpus from the format described by (Raganato et al., 2017) to UFSAC, and vice-versa.

The script **`ComputeMostFrequentSenses`** will calculate, for every lemma in WordNet, the most frequent sense (MFS), based on all usages in the given UFSAC corpora. This is helpful since in most evaluation campaigns, the MFS baseline (i.e. the score obtained when the MFS is assigned to every word) is important, and it is generally implicitly the sense distribution computed on the SemCor only.

The scripts **`AddCorpusLemma`** and **`AddCorpusPOS`** use respectively WordNet's *morphy* and Stanford's POS tagger to annotate a corpus with the lemma and POS of every word.

The script **`EvaluateWSD`** compare the sense annotations produced by a WSD system to the gold standard annotation, and compute the usual Precision, Recall, Coverage and F1 metrics for every given corpus.

The script **`GenerateCorpusStatistics`** is the one that was used to produce the table 1.

5. Experiments

In this section, we show an example of using all UFSAC corpora for the extension of a knowledge-based WSD system based on the Lesk measure. This experiment shows how this ressource can be used to easily improve an existing WSD system.

5.1. The Lesk and Extended Lesk Similarity Measures

(Lesk, 1986) proposed a simple algorithm for lexical disambiguation that evaluates the similarity between two senses (s_1, s_2) as the number of words in common in the definitions of the senses from a dictionary ($D(s_1)$,$D(s_2)$).
The Lesk measure compute an exact lexical match of the surface forms of the words in the definitions. If important words are missing or different synonyms of the same words are used in the definition of related senses, the overlap measure will not capture the proximity of their meanings appropriately. As definitions (especially in Princeton Wordnet) are very concise, it is difficult to obtain fine grained distinctions between senses.
In consequence, several variants of the Lesk measure tried to alleviate this problem, for instance (Baldwin et al., 2010) and (Miller et al., 2012), but the most common expansion technique is the so-called "extended/adapted Lesk" (Banerjee and Pedersen, 2002). The sense overlap is here expanded with the overlap of all definitions from all pairs of related senses in a lexico-semantic resource with a rich structure, such as WordNet.
In our experiment, we will create another expansion of the Lesk measure, based on UFSAC sense annotated corpora.

5.2. Expansion of Definitions Through UFSAC Sense-Annotated Corpora

Our method consists in expanding definitions with all neighbours of a target sense, taken from sense-annotated corpora. We consider that a neighbour is a word found in the same sentence as the target sense. More precisely, we proceed as following:

1. We parse every UFSAC corpus, sentence by sentence.
2. For every word which is sense-annotated in a sentence, we add to the definition of this sense in the dictionary every other word present in the sentence.

That is, for every sentence $S = w_0, w_1, \ldots, w_n$, and for every word w_k inside S, we add to the definition of the tagged sense of w_k, i.e. $D(s(w_k))$, every other words of the sentence, i.e. $w_i \forall i \in [0, n] i \neq k$.
As a consequence, every sense's definition in the dictionary will be extended with words that are related to this sense, in the same manner than (Banerjee and Pedersen, 2002)'s extended Lesk, but with words taken from sense annotated corpora.

5.3. Similarity-Based Word Sense Disambiguation

Now for evaluating this new expansion to the Lesk measure, we must use a similarity-based WSD algorithm that belongs to the broader category of knowledge-based approaches (using dictionaries, lexical base, encyclopedias...). In such systems, the disambiguation process consists of two layers: a local algorithm and a global algorithm. The local algorithm computes the proximity of two word senses, namely a semantic relatedness measure. The local similarity measurement is then used to find an optimal global sense assignment for all the content words of the text by the global algorithm. The local algorithm is here the Lesk measure augmented with the sense annotated corpora. We also filtered out stopwords according to the "long" list given in `https://www.ranks.nl/stopwords`.
As for global algorithms, they are often probabilistic combinatorial optimization algorithms, as WSD is fundamentally a discrete combinatorial optimization problem. Many such algorithms have been adapted to WSD, including genetic algorithms (Gelbukh et al., 2003), simulated annealing (Cowie et al., 1992), ant colony algorithms (Schwab et al., 2012) or more recently bee hive algorithms (Abualhaija and Zimmermann, 2016).
The different global algorithms mainly differ in the convergence speed to a close-to-optimal solution, however the bottleneck to the accuracy of the algorithm is the local algorithm (similarity measure) used, as it encodes the knowl-

System	SemEval 2007 Task 07	SemEval 2015 Task 13
Lesk + UFSAC corpora	79.83%	66.43%
Lesk	68.70%	50.65%
Extended Lesk (Banerjee and Pedersen, 2003)	78.01%	61.42%
Most Frequent Sense Baseline	78.90%	67.10%

Table 2: F1 scores of our similarity-based system augmented with words taken from all UFSAC corpora (except the evaluation corpora) on SemEval 2007 coarse-grained all-words task and SemEval 2015 fine-grained all-words task, compared to the Lesk, Extended Lesk and MFS baselines.

edge from the resource that allows to discriminate between the senses.

In this experiments, we use an adaptation to WSD of the Cuckoo Search Algorithm, the state of the art in combinatorial search algorithms (Yang and Deb, 2009). The algorithm relies on the Lévy flight distribution for an effective (and more meaningful) sampling of the search space.

The Cuckoo Search Algorithm is probabilistic and its result differs slightly from an execution to another (by an order of magnitude of less than 1%). So for each experiment, 30 executions are performed. Then, using a Shapiro-Wilk test (Shapiro and Wilk, 1965), we determined that none of the result distribution follow a normal distribution. Thus, we used a non-parametric Wilcoxon/Mann-Whitney-U (Wilcoxon, 1945) (Mann and Whitney, 1947) test in order to check the pairwise significance ($p < 0.01$) of all pairs of result distributions.

5.4. Results

We evaluate the performance of our expansion of definitions using all UFSAC corpora listed in subsection 4.1. except the ones we evaluated our system on: SemEval 2007 task 7 and SemEval 2015 task 13. We compare our similarity measure to the original Lesk and the Extended Lesk (Banerjee and Pedersen, 2003) measures. The results are presented in Table 2.

As we can see, our expansion of the definitions with words taken from the UFSAC corpora improves considerably the original Lesk measure, even more than the Extended Lesk measure. Therefore, this experiment demonstrates how much the addition of the UFSAC resource can improve a similarity-based WSD sytem. Of course, every other kind of WSD system can be improved, in particular supervised systems which rely solely on sense-annotated corpora and machine learning techniques (SVM, neural networks, etc.).

6. Conclusion

In this paper we advocate for a more uniform way of distributing sense annotated corpora, through a unique and uncomplicated file format. This unification can facilitate both the creation and the evaluation of Word Sense Disambiguation systems. Indeed, sense annotated corpora are historically separated between those created for the purpose of training, and those created for the purpose of evaluation. In addition, the formats of these corpora are often very different from each other: different file hierarchy, different syntax, and different sense inventory are used. Consequently, most WSD systems are trained and evaluated on few corpora comparing to the amount of existing corpora. Moreover, they are systematically evaluated only on corpora originally created for the purpose of evaluation, and trained only on corpora originally created for the purpose of training, whereas they could benefit from considering all of them in both tasks.

The unification of all sense annotated corpora hence allows to quickly expand a system which is trained on some resources to new data without the effort of writing another parser. Also, a system can now easily include to its training phase some corpora that were originally created for evaluation, and/or evaluate its performance on parts of corpora originally created for training. This easily allows a much better coverage and a more fine-grained analysis of a WSD system performance.

In our language resource, we gathered all existing English sense annotated corpora that we know, and we converted them in a simple and consistent XML file format that we named UFSAC. We also converted their sense annotations to the last version of WordNet (3.0). The corpora are only available when the licence authorizes it, but we also provide scripts that can easily convert a corpus from its original format to the one we propose. Thus, anyone who possess the corpora that we cannot distribute can still benefit from this work. In addition, we provide a complete Java API for reading, writing and modifying corpora in our unified format, along with example codes and tools for many applications such as lemmatization, POS-tagging, sense distribution estimation, etc. Finally, a demonstration of a simple use of all UFSAC corpora for extending a similarity-based WSD system is shown in section 5..

In the future, we plan to add to our resource other corpora such as the corpora created for the lexical sample tasks of SensEval/SemEval, and sense annotated corpora in other languages. We also plan to improve the UFSAC format by adding a better support for multiword expressions. The resource will be continuously updated at this url: `https://github.com/getalp/UFSAC`.

7. Bibliographical References

Abualhaija, S. and Zimmermann, K.-H. (2016). D-bees: A novel method inspired by bee colony optimization for solving word sense disambiguation. *Swarm and Evolutionary Computation*, pages –.

Baldwin, T., Kim, S., Bond, F., Fujita, S., Martinez, D., and Tanaka, T. (2010). A reexamination of mrd-based word sense disambiguation. 9(1):4:1–4:21, March.

Banerjee, S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing 2002*, Mexico City, February.

Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810.

Burnard, L. (1998). *The British National Corpus*.

Chan, Y. S., Ng, H. T., and Zhong, Z. (2007). Nus-pt: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 253–256, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chen, X., Liu, Z., and Sun, M. (2014). A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar, October. Association for Computational Linguistics.

Cowie, J., Guthrie, J., and Guthrie, L. (1992). Lexical disambiguation using simulated annealing. In *COLING 1992*, volume 1, pages 359–365, Nantes, France, août.

Daudé, J., Padró, L., and Rigau, G. (2000). Mapping wordnets using structural information. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 504–511, Stroudsburg, PA, USA. Association for Computational Linguistics.

Francis, W. N. and Kučera, H. (1964). A standard corpus of present-day edited american english, for use with digital computers (brown). Technical report, Brown University, Providence, Rhode Island.

Gelbukh, A., Sidorov, G., and Han, S. Y. (2003). Evolutionary approach to natural language wsd through global coherence optimization. *WSEAS Transactions on Communications*, 2(1):11–19.

Habert, B., Fabre, C., and Issac, F. (1998). *DE L'ECRIT AU NUMERIQUE. Constituer, normaliser et exploiter les corpus électroniques*. Number ISBN : 2-225-82953-5. ELSEVIER MASSON.

Ide, N. and Macleod, C. (2001). The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics 2001*, volume 3.

Lesk, M. (1986). Automatic sense disambiguation using mrd: how to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC '86*, pages 24–26, New York, NY, USA. ACM.

Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 303–308, Stroudsburg, PA, USA. Association for Computational Linguistics.

Miller, T., Biemann, C., Zesch, T., and Gurevych, I. (2012). Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proceedings of COLING 2012*, pages 1781–1796, Mumbai, India, December. The COLING 2012 Organizing Committee.

Miller, G. A. (1995). Wordnet: A lexical database. *ACM*, Vol. 38(No. 11):p. 1–41.

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

Màrquez, L., Raya, J., Carroll, J., McCarthy, D., Agirre, E., Martínez, D., Strapparava, C., and Gliozzo, A. (2002). Experiment a : Several all-words wsd systems for english. Technical report, Meaning, Developing multilingual Web-scale Language Technologies.

Navigli, R. and Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics.

Navigli, R., Litkowski, K. C., and Hargraves, O. (2007). Semeval-2007 task 07: Coarse-grained english all-words task. In *SemEval-2007*, pages 30–35, Prague, Czech Republic, June.

Navigli, R. (2012). A quick tour of word sense disambiguation, induction and related approaches. In *Proceedings of the 38th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 115–129.

Raganato, A., Camacho-Collados, J., and Navigli, R. (2017). Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain, April. Association for Computational Linguistics.

Schwab, D., Goulian, J., Tchechmedjiev, A., and Blanchon, H. (2012). Ant Colony Algorithm for the Unsupervised Word Sense Disambiguation of Texts: Comparison and Evaluation. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2012)*, Mumbai (India), dec.

Schwab, D., Goulian, J., and Tchechmedjiev, A. (2013). Désambiguïsation lexicale de textes : efficacité qualitative et temporelle d'un algorithme à colonies de fourmis. *TAL*, 54(1):99–138.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 3(52).

Taghipour, K. and Ng, H. T. (2015). One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing, China, July. Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Con-*

ference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vial, L., Tchechmedjiev, A., and Schwab, D. (2016). Extension lexicale de définitions grâce à des corpus annotés en sens. In *Traitement Automatique des Langues Naturelles (TALN)*.

Vial, L., Lecouteux, B., and Schwab, D. (2017). Uniformisation de corpus anglais annotés en sens. In *24ème Conférence sur le Traitement Automatique des Langues Naturelles*, Orléans, France, June.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, December.

Yang, X.-S. and Deb, S. (2009). Cuckoo search via lévy flights. *Proc. of World Congress on Nature and Biologically Inspired Computing*, pages 210–214.

Yuan, D., Richardson, J., Doherty, R., Evans, C., and Altendorf, E. (2016). Semi-supervised word sense disambiguation with neural models. In *COLING 2016*.

8. Language Resource References

Hovy et al. (2006). *OntoNotes: The 90% Solution*.

Ide et al. (2008). *MASC: the Manually Annotated Sub-Corpus of American English*.

Miller et al. (1993). *A Semantic Concordance*.

Miller. (1995). *Wordnet: A Lexical Database*.

Ng and Lee. (1997). *DSO Corpus of Sense-Tagged English*.

Taghipour and Ng. (2015). *One Million Sense-Tagged Instances for Word Sense Disambiguation and Induction*.