

# Werdy: Recognition and Disambiguation of Verbs and Verb Phrases with Syntactic and Semantic Pruning

Luciano Del Corro

Rainer Gemulla

Gerhard Weikum

Max-Planck-Institut für Informatik  
Saarbrücken, Germany

{ delcorro, rgemulla, weikum }@mpi-inf.mpg.de

## Abstract

Word-sense recognition and disambiguation (WERD) is the task of identifying word phrases and their senses in natural language text. Though it is well understood how to disambiguate noun phrases, this task is much less studied for verbs and verbal phrases. We present Werdy, a framework for WERD with particular focus on verbs and verbal phrases. Our framework first identifies multi-word expressions based on the syntactic structure of the sentence; this allows us to recognize both contiguous and non-contiguous phrases. We then generate a list of candidate senses for each word or phrase, using novel syntactic and semantic pruning techniques. We also construct and leverage a new resource of pairs of senses for verbs and their object arguments. Finally, we feed the so-obtained candidate senses into standard word-sense disambiguation (WSD) methods, and boost their precision and recall. Our experiments indicate that Werdy significantly increases the performance of existing WSD methods.

## 1 Introduction

Understanding the semantics of words and multi-word expressions in natural language text is an important task for automatic knowledge acquisition. It serves as a fundamental building block in a wide area of applications, including semantic parsing, question answering, paraphrasing, knowledge base construction, etc. In this paper, we study the task of word-sense recognition and disambiguation (WERD) with a focus on verbs and

verbal phrases. Verbs are the central element in a sentence, and the key to understand the relations between sets of entities expressed in a sentence.

We propose Werdy, a method to (i) automatically recognize in natural language text both single words and multi-word phrases that match entries in a lexical knowledge base (KB) like WordNet (Fellbaum, 1998), and (ii) disambiguate these words or phrases by identifying their senses in the KB. WordNet is a comprehensive lexical resource for word-sense disambiguation (WSD), covering nouns, verbs, adjectives, adverbs, and many multi-word expressions. In the following, the notion of an *entry* refers to a word or phrase in the KB, whereas a *sense* denotes the lexical synset of the entry's meaning in the given sentence.

A key challenge for recognizing KB entries in natural language text is that entries often consist of multiple words. In WordNet-3.0 more than 40% of the entries are multi-word. Such entries are challenging to recognize accurately for two main reasons: First, the components of multi-word entries in the KB (such as *fiscal year*) often consist of components that are themselves KB entries (*fiscal* and *year*). Second, multi-word entries (such as *take a breath*) may not appear consecutively in a sentence (“He takes a deep breath.”). Werdy addresses the latter problem by (conceptually) matching the syntactic structure of the KB entries to the syntactic structure of the input sentence. To address the former problem, Werdy identifies all possible entries in a sentence and passes them to the disambiguation phase (*take, breath, take a breath, ...*); the disambiguation phase provides more information about which multi-word entries to keep. Thus, our method solves the recognition and the disambiguation tasks jointly.

Once KB entries have been identified, Werdy

disambiguates each entry against its possible senses. State-of-the-art methods for WSD (Navigli, 2009) work fairly well for nouns and noun phrases. However, the disambiguation of verbs and verbal phrases has received much less attention in the literature.

WSD methods can be roughly categorized into (i) methods that are based on supervised training over sense-annotated corpora (e.g., Zhong and Ng (2010)), and (ii) methods that harness KB's to assess the semantic relatedness among word senses for mapping entries to senses (e.g., Ponzetto and Navigli (2010)). For these methods, mapping verbs to senses is a difficult task since verbs tend to have more senses than nouns. In WordNet, including monosemous words, there are on average 1.24 senses per noun and 2.17 per verb.

To disambiguate verbs and verbal phrases, Werdy proceeds in multiple steps. First, Werdy obtains the set of candidate senses for each recognized entry from the KB. Second, it reduces the set of candidate entries using novel syntactic and semantic pruning techniques. The key insight behind our syntactic pruning is that each verb sense tends to occur in a only limited number of *syntactic patterns*. For example, the sentence “Albert Einstein remained in Princeton” has a subject (“Albert Einstein”), a verb (“remained”) and an adverbial (“in Princeton”), it follows an SVA clause pattern. We can thus safely prune verb senses that do not match the syntactic structure of the sentence. Moreover, each verb sense is compatible with only a limited number of *semantic argument types* (such as *location*, *river*, *person*, *musician*, etc); this phenomena is called *selectional preference* or *selectional restriction*. Senses that are compatible only with argument types not present in the sentence can be pruned. Our pruning steps are based on the idea that a verb selects the categories of its arguments both syntactically (c-selection) and semantically (s-selection). In the final step, Werdy employs a state-of-the-art general WSD method to select the most suitable sense from the remaining candidates. Since incorrect senses have already been greatly pruned, this step significantly gains accuracy and efficiency over standard WSD.

Our semantic pruning technique builds on a newly created resource of pairs of senses for verbs and their object arguments. For example, the WordNet verb sense  $\langle play-1 \rangle$  (i.e., the 1st sense of

the verb entry “play”) selects as direct object the noun sense  $\langle sport-1 \rangle$ . We refer to this novel resource as the *VO Sense Repository*, or VOS repository for short.<sup>1</sup> It is constructed from the WordNet gloss-tags corpus, the SemCor dataset, and a small set of manually created VO sense pairs.

We evaluated Werdy on the SemEval-2007 coarse-grained WSD task (Navigli et al., 2007), both with and without automatic recognition of entries. We found that our techniques boost state-of-the-art WSD methods and obtain high-quality results. Werdy significantly increases the precision and recall of the best performing baselines.

The rest of the paper is organized as follows. Section 2 gives an overview of Werdy components. Section 3 presents the entry recognition, and Sections 4 and 5 discuss our novel syntactic and semantic pruning techniques. Section 6 presents the Semantic VO Repository and how we constructed it. Section 7 gives the results of our evaluation. Section 8 discusses related work.

## 2 Overview of Werdy

Werdy consists of four steps: (i) entry recognition, (ii) syntactic pruning, (iii) semantic pruning, and (iv) word-sense disambiguation. The novel contribution of this paper is in the first three steps, and in the construction of the VO sense repository. Each of these steps operates on the clause level, i.e., we first determine the set of clauses present in the input sentence and then process clauses separately. A *clause* is a part of a sentence that expresses some statement or coherent piece of information. Clauses are thus suitable minimal units for automatic text understanding tasks (Del Corro and Gemulla, 2013); see Sec.3 for details.

In the *entry-recognition step* (Sec. 3), Werdy obtains for the input sentence a set of potential KB entries along with their part-of-speech tags. The candidate senses of each entry are obtained from WordNet. For instance, in the sentence “He takes a deep and long breath”, the set of potential entries includes *take* (verb, 44 candidate senses), *take a breath* (verb, 1 candidate sense), and *breath* (noun, 5 candidate senses). Note that in contrast to Werdy, most existing word-sense disambiguation methods assume that entries have already been (correctly) identified.

---

<sup>1</sup>The VOS repository, Werdy’s source code, and results of our experimental study are available at <http://people.mpi-inf.mpg.de/~corrogg/>.

In the *syntactic-pruning step* (Sec. 4), we eliminate candidate senses that do not agree with the syntactic structure of the clause. It is well-established that the syntactic realization of a clause is intrinsically related with the sense of its verb (Quirk et al., 1985; Levin, 1993; Hanks, 1996; Baker et al., 1998; Palmer et al., 2005). Quirk et al. (1985) identified seven possible clause types in the English language (such as “subject verb adverbial”, SVA). We make use of techniques inspired by Del Corro and Gemulla (2013) to identify the clause type of each clause in the sentence. We then match the clause type with the set of WordNet frames (e.g., “somebody verb something”) that WordNet provides for each verb sense, and prune verb senses for which there is no match.

In the *semantic-pruning step* (Sec. 5), we further prune the set of candidate senses by taking the semantic types of direct objects into account. Similarly to the syntactic relation mentioned above, a verb sense also imposes a (selectional) restriction on the semantic type of its arguments (Quirk et al., 1985; Levin, 1993; Hanks, 1996; Baker et al., 1998; Palmer et al., 2005). For instance, the verb *play* with sense *participate in games or sports* requires an object argument of type  $\langle game-1 \rangle^2$ ,  $\langle game-3 \rangle$ , or  $\langle sport-1 \rangle$ . Senses that do not match the arguments found in the clause are pruned. This step is based on the newly constructed VOS Repository (Sec. 6). Note that when there is no direct object, only the syntactic pruning step applies.

### 3 Entry Recognition

The key challenge in recognizing lexical KB entries in text is that entries are not restricted to single words. In addition to named entities (such as people, places, etc.), KB’s contain multi-word expressions. For example, WordNet-3.0 contains entries such as *take place* (verb), *let down* (verb), *take into account* (verb), *be born* (verb), *high school* (noun), *fiscal year* (noun), and *Prime Minister* (noun). Note that each individual word in a multi-word entry is usually also an entry by itself, and can even be part of several multi-word entries. To ensure correct disambiguation, all potential multi-word entries need to be recognized (Finlayson and Kulkarni, 2011), even when they do not appear as consecutive words in a sentence.

Werdy addresses these challenges by exploring the syntactic structure of both the input sen-

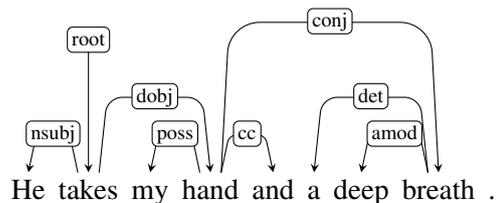


Figure 1: An example dependency parse

tence and the lexical KB entries. The structure of the sentence is captured in a dependency parse (DP). Given a word in a sentence, Werdy conceptually generates all subtrees of the DP starting at that word, and matches them against the KB. This process can be performed efficiently as WordNet entries are short and can be indexed appropriately. To match the individual words of a sentence against the words of a KB entry, we follow the standard approach and perform lemmatization and stemming (Finlayson, 2014). To further handle personal pronouns and possessives, we follow Arranz et al. (2005) and normalize personal pronouns (I, you, my, your, ...) to *one*’s, and reflexive pronouns (myself, yourself, ...) to *oneself*.

Consider the example sentence “He takes my hand and a deep breath”. We first identify the clauses and their DP’s (Fig. 1) using the method of Del Corro and Gemulla (2013), which also processes coordinating conjunctions. We obtain clauses “He takes my hand” and “He takes a deep breath”, which we process separately. To obtain possible entries for the first clause, we start with its head word (*take*) and incrementally consider its descendants (*take hand*, *take one’s hand*, ...). The exploration is terminated as early as possible; for example, we do not consider *take one’s hand* because there is no WordNet entry that contains both *take* and *hand*. For the second clause, we start with *take* (found in WordNet), then expand to *take breath* (not found but can occur together), then *take a breath* (found), then *take a deep breath* (not found, cannot occur together) and so on.

Note that the word “take” in the sentence refer to two different entries and senses: “take” for the first clause and “take a breath” for the second clause. In this stage no decisions are made about selecting entries and disambiguating them; these decisions are made in the final WSD stage of Werdy.

We tested Werdy’s entry-recognizer on the SemEval-2007 corpus. We detected the correct en-

<sup>2</sup>We use the notation  $\langle \text{WordNet entry-sense number} \rangle$ .

Pattern	Clause type	Example	WN frame example [frame number]
SV <sub>i</sub>	SV	AE died.	Somebody verb [2]
SV <sub>e</sub> A	SVA	AE remained in Princeton.	Somebody verb PP [22]
SV <sub>c</sub> C	SVC	AE is smart.	Somebody verb adjective [6]
SV <sub>mt</sub> O	SVO	AE has won the Nobel Prize.	Somebody verb something [8]
SV <sub>dt</sub> O <sub>i</sub> O	SVOO	RSAS gave AE the Nobel Prize.	Somebody verb somebody something [14]
SV <sub>ct</sub> OA	SVOA	The doorman showed AE to his office.	Somebody verb somebody PP [20]
SV <sub>ct</sub> OC	SVOC	AE declared the meeting open.	Something verb something adjective/noun [5]

S: Subject, V: Verb, C: Complement, O: Direct object, O<sub>i</sub>: Indirect object, A: Adverbial, V<sub>i</sub>: Intransitive verb, V<sub>c</sub>: Copular verb,  
V<sub>e</sub>: Extended-copular verb, V<sub>mt</sub>: Monotransitive verb, V<sub>dt</sub>: Ditransitive verb, V<sub>ct</sub>: Complex-transitive verb

Table 1: Clause types and examples of matching WordNet frames

tries for all but two verbs (out of more than 400). The two missed entries (*take up* and *get rolling*) resulted from incorrect dependency parses.

#### 4 Syntactic Pruning

Once the KB entries have been recognized, Werdy prunes the set of possible senses of each verb entry by considering the syntactic structure of the clause in which the entry occurs. This pruning is based on the observation that each verb sense may occur only in a limited number of “clause types”, each having specific semantic functions (Quirk et al., 1985). When the clause type of the sentence is incompatible with a candidate sense of an entry, this sense is eliminated.

Werdy first detects in the input sentence the set of clauses and their constituents. A clause consists of one *subject* (S), one *verb* (V), and optionally an *indirect object* (O), a *direct object* (O), a *complement* (C) and one or more *adverbials* (A). Not all combinations of clause constituents appear in the English language. When we classify clauses according to the grammatical function of their constituents, we obtain only seven different clause types (Quirk et al., 1985); see Table 1. For example, the sentence “He takes my hand” is of type SVO; here “He” is the subject, “takes” the verb, and “my hand” the object. The clause type can (in principle) be determined by observing the verb type and its complementation (Del Corro and Gemulla, 2013).

For instance, consider the SVA clause “The student remained in Princeton”. The verb *remain* has four senses in WN: (1) stay the same; remain in a certain state (e.g., “The dress remained wet”), (2) continue in a place, position, or situation (“He remained dean for another year”), (3) be left; of persons, questions, problems (“There remains the

question of who pulled the trigger”) or (4) stay behind (“The hostility remained long after they made up”). The first sense of *remain* requires an SVC pattern; the other cases require either SV or SVA. Our example clause is of type SVA so that we can safely prune the first sense.

WordNet provides an important resource for obtaining the set of clause types that are compatible with each sense of a verb. In particular, each verb sense in WordNet is annotated with a set of *frames* (e.g., “somebody verb something”) in which they may occur, capturing both syntactic and semantic constraints. There are 35 different frames in total.<sup>3</sup> We manually assigned a set of clause types to each frame (e.g., SVO to frame “somebody verb something”). Table 1 shows an example frame for each of the seven clause types. On average, each WordNet-3.0 verb sense is associated with 1.57 frames; the maximum number of frames per sense is 9. The distribution of frames is highly skewed: More than 61% of the 21,649 frame annotations belong to one of four simple SVO frames (numbers 8, 9, 10 and 11), and 22 out of the 35 frames have less than 100 instances. This skew makes the syntactic pruning step effective for non-SVO clauses, but less effective for SVO clauses.

Werdy directly determines a set of possible frame types for each clause of the input sentence. Our approach is based on the clause-type detection method of Del Corro and Gemulla (2013), but we also consider additional information that is captured in frames but not in clause types. For example, we distinguish different realizations of objects (such as clausal objects from non-clausal objects), which are not captured in the clause type. Given the DP of a clause, Werdy identifies the

<sup>3</sup>See <http://wordnet.princeton.edu/wordnet/man/wninput.5WN.html>.

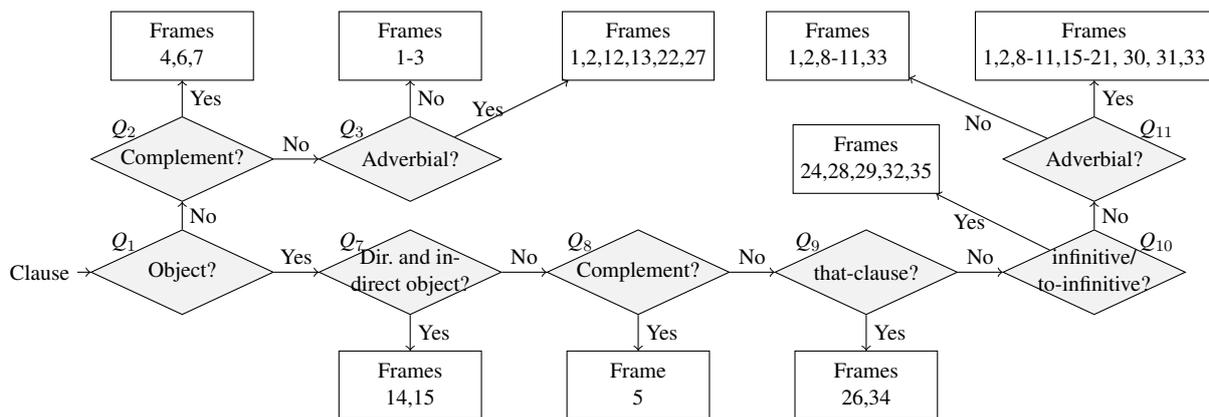


Figure 2: Flow chart for frame detection

set of WN frames that can potentially match the clause as outlined in the flowchart of Fig. 2. Werdy walks through the flowchart; for each question, we check for the presence or absence of a specific constituent of a clause (e.g., a direct object for  $Q_1$ ) and proceed appropriately until we obtain a set of possible frames. This set is further reduced by considering additional information in the frames (not shown; e.g., that the verb must end on “-ing”). For our example clause “The student remained in Princeton”, we first identify possible frames  $\{1, 2, 12, 13, 22, 27\}$  using the flowchart ( $Q_1$  no,  $Q_2$  no,  $Q_3$  yes); using the additional information in the frames, Werdy then further prunes this set to  $\{1, 2, 22\}$ . The corresponding set of remaining candidate sense for *remain* is as given above, i.e.,  $\{\langle remain-2 \rangle, \langle remain-3 \rangle, \langle remain-4 \rangle\}$ .

Our mapping of clause types to WordNet frames is judiciously designed for the way WordNet is organized. For instance, frames containing adverbials generally do not specify whether or not the adverbial is obligatory; here we are conservative in that we do not prune such frames if the input clause does not contain an adverbial. As another example, some frames overlap or subsume each other; e.g, frame “somebody verb something” (8) subsumes “somebody verb that clause” (26). In some word senses annotated with the more general frame, the more specific one can also apply (e.g.,  $\langle point\ out-1 \rangle$  is annotated with 8 but not 26; 26 can apply), in others it does not (e.g.,  $\langle play-1 \rangle$  is also annotated with 8 but not 26; but here 26 cannot apply). To ensure the effectiveness of syntactic pruning, we only consider the frames that are directly specified in WordNet. This procedure often produces the desired results; in a few cases, however, we do prune the correct sense (e.g., frame 26

for clause “He points out that . . .”).

## 5 Semantic Pruning

A verb sense imposes a restriction on the semantic type of the arguments it may take and vice versa (Quirk et al., 1985; Levin, 1993; Hanks, 1996; Baker et al., 1998; Palmer et al., 2005; Kipper et al., 2008). This allows us to further prune the verb candidate set by discarding verb senses whose semantic argument is not present in the clause.

WordNet frames potentially allow a shallow type pruning based on the semantics provided for the clause constituents. However we could solely distinguish people (“somebody”) from things (“something”), which is too crude to obtain substantial pruning effects. Moreover, this distinction is sometimes ambiguous.

Instead, we have developed a more powerful approach to semantic pruning based on our VOS repository. We remove from the verb candidate set those senses whose semantic argument cannot be present in the sentence. For instance, consider the clause “The man plays football.” Suppose that we know that the verb entry *play* with sense  $\langle play-1 \rangle$  (“participate in sports”) takes an object of type  $\langle sport-1 \rangle$ ; i.e., we have a tuple  $\langle play-1, sport-1 \rangle$  in our repository. Then, we check whether any of the possible senses of *football*—(i) *sport* or (ii) *ball*—is of type  $\langle sport-1 \rangle$ . Here the first sense has the correct type (the second sense does not); thus we retain  $\langle play-1 \rangle$  as a possible sense for the verb entry *play*. Next, suppose that we consider sense  $\langle play-3 \rangle$  (“play on an instrument”), which according to our corpus takes  $\langle instrument-6 \rangle$  as argument (i.e., there is a tuple  $\langle play-3, instrument-6 \rangle$  in our VOS repository). Since none of the senses of *football* is of type  $\langle instrument-6 \rangle$ , we can safely drop

$\langle play-3 \rangle$  from our candidate set. We perform this procedure for every verb sense in the candidate set.

Semantic pruning makes use of both VOS repository and the hypernym structure of the noun senses in WordNet. For each sentence, we obtain the possible senses of the direct-object argument of the verb. We then consider each candidate sense of the verb (e.g.,  $\langle play-1 \rangle$ ), and check whether any of its compatible object-argument senses (from our repository) is a hypernym of any of the possible senses of its actual object argument (in the sentence); e.g.,  $\langle sport-1 \rangle$  is a hypernym of  $\langle football-1 \rangle$ . If so, we retain the verb’s candidate sense. If not, either the candidate sense of the verb is indeed incompatible with the object argument in the sentence, or our repository is incomplete. To handle incompleteness to some extent, we also consider hyponyms of the object-argument senses in our repository; e.g., if we observe object *sport* in a sentence and have verb-sense argument  $\langle football-1 \rangle$  in our corpus, we consider this a match. If the hyponyms lead to a match, we retain the verb’s candidate sense; otherwise, we discard it.

## 6 Verb-Object Sense Repository

We use three different methods to construct the repository. In particular, we harness the sense-annotated WordNet glosses<sup>4</sup> as well as the sense-annotated SemCor corpus (Landes et al., 1998).<sup>5</sup>

The major part of the VOS repository was acquired from WordNet’s gloss tags. According to Atkins and Rundell (2008), noun definitions should be expressed in terms of the class to which they belong, and verb definitions should refer to the types of the subjects or objects related to the action. Based on this rationale, we extracted all noun senses that appear in the gloss of each verb sense; each of these noun senses is treated as a possible sense of the object argument of the corresponding verb sense. For example, the gloss of  $\langle play-1 \rangle$  is “participate in games or sports;” each noun is annotated with its senses (2 and 3 for “games”, 1 for “sports”). We extract tuples  $\langle play-1, game-2 \rangle$ ,  $\langle play-1, game-3 \rangle$ , and  $\langle play-1, sport-1 \rangle$  from this gloss. Note that we only extract direct-object arguments, i.e., we do not consider the type of the subject argument of a verb sense. Since the constituents of the predicate are much

more important than the subject to determine or describe a verb sense, lexical resources rarely contain information on the subject (Atkins and Rundell, 2008). Similarly, WordNet glosses typically do not provide any information about adverbials. Overall, we collected arguments for 8,657 verb senses (out of WordNet’s 13,767 verb senses) and a total of 13,050  $\langle verb-\#, object-\# \rangle$ -pairs.

We leveraged the sense-annotated SemCor corpus to further extend our VOS repository. We parsed each sentence in the corpus to obtain the respective pairs of verb sense and object sense. Since sentences are often more specific than glosses, and thus less helpful for constructing our repository, we generalized the so-found object senses using a heuristic method. In particular, we first obtained all the object senses of each verb sense, and then repeatedly generalized sets of *at least two senses* that share a direct hypernym to this hypernym. The rationale is that we only want to generalize if we have some evidence that a more general sense may apply; we thus require at least two hyponyms before we generalize. Using this method, we collected arguments for 1,516 verb senses and a total of 4,131 sense pairs.

Finally, we noticed that the most frequent senses used in the English language are usually so general that their glosses do not contain any relevant semantic argument. For instance, one of the most frequent verbs is  $\langle see-1 \rangle$ , which has gloss “perceive by  $\langle sight-3 \rangle$ ”. The correct semantic argument  $\langle entity-1 \rangle$  is so general that it is omitted from the gloss. In fact, our gloss-tag extractor generates tuple  $\langle see-1, sight-3 \rangle$ , which is incorrect. We thus manually annotated the 30 most frequent verb senses with their object argument types.

Our final repository contains arguments for 9,335 verb senses and a total of 17,181 pairs. Pairs from SemCor tend to be more specific because they refer to text occurrences. The assumption of taking the nouns of the glosses as arguments seems to be mostly correct, although some errors may be introduced. Consider the pair  $\langle play-28, stream-2 \rangle$  extracted from the gloss “discharge or direct or be discharged or directed as if in a continuous  $\langle stream-2 \rangle$ ”. Also, in some cases, the glosses may refer to adverbials as in  $\langle play-14, location-1 \rangle$ , taken from gloss “perform on a certain  $\langle location-1 \rangle$ ”. Note that if an argument is missing from our repository, we may prune the correct sense of the verb. If, however, there is an additional, incorrect

<sup>4</sup><http://wordnet.princeton.edu/glosstag.shtml>

<sup>5</sup><http://web.eecs.umich.edu/~mihalcea/downloads.html>

argument in the repository, the correct verb sense is retained but pruning may be less effective.

## 7 Evaluation

**Dataset.** We tested Werdy on the SemEval-2007 coarse-grained dataset.<sup>6</sup> It consists of five sense-annotated documents; the sense annotations refer to a coarse-grained version of WordNet. In addition to sense annotations, the corpus also provides the corresponding KB entries (henceforth termed “gold entries”) as well as a POS tag. We restrict our evaluation to verbs that act as clause heads. In total, 461 such verbs were recognized by ClausIE (Del Corro and Gemulla, 2013) and the Stanford Parser (Klein and Manning, 2003).<sup>7</sup>

**WSD Algorithms.** For the final step of Werdy, we used the KB-based WSD algorithms of Ponzetto and Navigli (2010) and It-Makes-Sense (Zhong and Ng, 2010), a state-of-the-art supervised system that was the best performer in SemEval-2007. Each method only labels entries for which it is sufficiently confident.

*Simplified Extended Lesk (SimpleExtLesk).* A version of Lesk (1986). Each entry is assigned the sense with highest term overlap between the entry’s context (words in the sentence) and both the sense’s gloss (Kilgarriff and Rosenzweig, 2000) as well as the glosses of its neighbors (Banerjee and Pedersen, 2003). A sense is output only if the overlap exceeds some threshold; we used thresholds in the range of 1–20 in our experiments. There are many subtleties and details in the implementation of SimpleExtLesk so we used two different libraries: a Java implementation of WordNet::Similarity (Pedersen et al., 2004),<sup>8</sup> which we modified to accept a context string, and DKPro-WSD (Miller et al., 2013) version 1.1.0, with lemmatization, removal of stop words, paired overlap enabled and normalization disabled.

*Degree Centrality.* Proposed by Navigli and Lapata (2010). The method collects all paths connecting each candidate sense of an entry to the set of candidate senses of the words the entry’s context. The candidate sense with the highest degree in the resulting subgraph is selected. We implemented this algorithm using the Neo4j library.<sup>9</sup>

<sup>6</sup>The data is annotated with WordNet 2.1 senses; we converted the annotations to WordNet-3.0 using DKPro-WSD (Miller et al., 2013).

<sup>7</sup>Version 3.3.1, model englishRNN.ser.gz

<sup>8</sup><http://www.sussex.ac.uk/Users/drh21/>

<sup>9</sup><http://www.neo4j.org/>

We used a fixed threshold of 1 and vary the search depth in range 1–20. We used the candidate senses of all nouns and verbs in a sentence as context.

*It-Makes-Sense (IMS).* A state-of-the-art, publicly available supervised system (Zhong and Ng, 2010) and a refined version of Chan et al. (2007), which ranked first in the SemEval-2007 coarse grained task. We modified the code to accept KB entries and their candidate senses. We tested both in WordNet-2.1 and 3.0; for the later we mapped Werdy’s set of candidates to WordNet-2.1.

*Most Frequent Sense (MFS).* Selects the most frequent sense (according to WordNet frequencies) among the set of candidate senses of an entry. If there is a tie, we do not label. Note that this procedure differs slightly from the standard of picking the entry with the smallest sense id. We do not follow this approach since it cannot handle well overlapping entries.

*MFS back-off.* When one of the above methods fails to provide a sense label (or provides more than one), we used the MFS method above with a threshold of 1. This procedure increased the performance in all cases.

**Methodology.** The disambiguation was performed with respect to coarse-grained sense clusters. The score of a cluster is the sum of the individual scores of its senses (except for IMS which provides only one answer per word); the cluster with the highest score was selected. Our source code and the results of our evaluation are publicly available<sup>10</sup>.

The SemEval-2007 task was not designed for automatic entry recognition, for each word or multi-word expression it provides the WordNet entry and the POS tag. We proceeded as follows to handle multi-word entries. In the WSD step, we considered the candidate senses of all recognized entries that overlap with the gold entry. For example, we considered the candidate senses of entries *take*, *breath*, and *take a breath* for gold entry *take a breath*.

The SemEval-2007 task uses WordNet-2.1 but Werdy uses WordNet-3.0. We mapped both the sense keys and clusters from WordNet-2.1 to WordNet-3.0. All senses in WordNet-3.0 that could not be mapped to any cluster were considered to belong each of them to a single sense cluster. Note that this procedure is fair: for such senses

<sup>10</sup><http://people.mpi-inf.mpg.de/~corrogg/>

Algorithm	Gold Entry	Pruning	MFS back-off	threshold /depth	Verbs (clause heads)			F1 points	
					P	R	F1		
Degree Centrality	+	-	+	5	73.54	73.54	73.54		
	+	+	+	11	79.61	79.61	79.61	+ 6.07	
	+	-	-	5	73.99	71.58	72.77		
	+	+	-	8	79.91	78.52	79.21	+ 6.44	
	-	-	+	5	70.41	70.41	70.41		
	-	+	+	10	76.46	76.46	76.46	+ 6.05	
	-	-	-	4	71.05	68.90	69.96		
	-	+	-	10	76.81	75.81	76.30	+ 6.34	
	SimpleExtLesk (DKPro)	+	-	+	6	77.28	75.27	76.26	
		+	+	+	5	81.90	80.48	<b>81.18</b>	+ 4.92
+		-	-	1	73.70	52.28	61.17		
+		+	-	1	81.99	64.21	72.02	<b>+ 10.85</b>	
-		-	+	5	74.33	72.57	73.44		
-		+	+	5	79.30	77.75	<b>78.52</b>	+ 5.08	
-		-	-	1	69.85	50.54	58.65		
-		+	-	1	78.69	62.20	69.48	+ 10.83	
SimpleExtLesk (WordNet:Sim)		+	-	+	5	77.11	75.27	76.18	
		+	+	+	5	80.57	79.18	79.87	+ 3.69
	+	-	-	1	74.82	68.98	71.78		
	+	+	-	1	79.04	75.27	77.11	+ 5.33	
	-	-	+	6	74.12	72.35	73.22		
	-	+	+	7	77.97	76.46	77.21	+ 3.99	
	-	-	-	1	71.36	65.66	68.39		
	-	+	-	1	76.20	71.92	74.00	+ 5.61	
	MFS	+	-	-	1	76.61	74.62	75.60	
		+	+	-	1	80.35	78.96	79.65	+ 4.05
-		-	-	1	73.67	71.92	72.79		
-		+	-	1	77.75	76.24	76.99	+ 4.20	
IMS (WordNet-2.1)	+	-	+	n.a.	79.60	79.60	79.60		
	+	+	+	n.a.	80.04	80.04	80.04	+ 0.44	
	-	-	+	n.a.	76.21	75.05	75.63		
	-	+	+	n.a.	77.53	76.36	76.94	+ 1.31	
IMS (WordNet-3.0)	+	-	+	n.a.	78.96	78.96	78.96		
	+	+	+	n.a.	79.83	79.83	79.83	+ 0.87	
	-	-	+	n.a.	75.77	74.62	75.19		
	-	+	+	n.a.	77.53	76.36	76.94	+ 1.75	

Table 2: Results on SemEval-2007 coarse-grained (verbs as clause heads)

the disambiguation is equivalent to a fine-grained disambiguation, which is harder.

**Results.** Our results are displayed in Table 2. We ran each algorithm with the gold KB entries provided by in the dataset (+ in column “gold entry”) as well as the entries obtained by our method of Sec. 3 (-). We also enabled (+) and disabled (-) the pruning steps as well as the MFS back-off strategy. The highest F1 score was achieved by SimpleExtLesk (DKPro) with pruning and MFS back-off: 81.18 with gold entries and 78.52 with automatic entry recognition. In all cases, our syntactic and semantic pruning strategy increased performance (up to +10.85 F1 points). We next discuss the impact of the various steps of Werdy in

detail.

**Detailed Analysis.** Table 3 displays step-by-step results for DKPro’s SimpleExtLesk, for MFS, as well as SimpleExtLesk with MFS back-off, the best performing strategy. The table shows results when only some Werdy’s steps are used. We start from a direct use of the respective algorithm with the gold entries of SemEval-2007 after each horizontal line, and then successively add the Werdy steps indicated in the table.

When no gold entries were provided, performance dropped due to the increase of sense candidates for multi-word expressions, which include the possible senses of the expression itself as well as the senses of the entry’s parts that are them-

Steps Performed	threshold	P	R	F1	F1 points
<b>SimpleExtLesk (DKPro)</b>					
<b>Plain with gold entries</b>	<b>1</b>	<b>73.70</b>	<b>52.28</b>	<b>61.17</b>	
+ Entry Recognition	1	69.85	50.54	58.65	- 2.52
+ Syntactic Pruning	1	76.47	58.84	66.50	+ <b>7.85</b>
+ Semantic Pruning	1	78.69	62.20	69.48	+ 2.98
+ Entry Recognition	1	69.85	50.54	58.65	- 2.52
+ Semantic Pruning	1	73.85	55.39	63.30	+ 4.65
+ Syntactic Pruning	1	79.33	61.21	69.10	+ <b>7.93</b>
+ Semantic Pruning	1	81.99	64.21	72.02	+ 2.92
+ Semantic Pruning	1	78.11	56.90	65.84	+ 4.67
<b>MFS</b>					
<b>Plain with gold entries</b>	<b>1</b>	<b>76.61</b>	<b>74.62</b>	<b>75.60</b>	
+ Entry Recognition	1	73.67	71.92	72.79	- 2.81
+ Syntactic Pruning	1	75.77	74.14	74.95	+ 2.16
+ Semantic Pruning	1	77.75	76.24	76.99	+ 2.04
+ Entry Recognition	1	73.67	71.92	72.79	- 2.81
+ Semantic Pruning	1	77.09	75.43	76.25	+ <b>3.46</b>
+ Syntactic Pruning	1	78.46	76.94	77.69	+ 2.09
+ Semantic Pruning	1	80.35	78.96	79.65	+ 1.96
+ Semantic Pruning	1	79.91	78.02	78.95	+ <b>3.35</b>
<b>SimpleExtLesk (DKPro) with MFS back-off</b>					
<b>Plain with gold entries</b>	<b>6</b>	<b>77.28</b>	<b>75.27</b>	<b>76.26</b>	
+ Entry Recognition	6	74.33	72.57	73.44	- 2.82
+ Syntactic Pruning	5	76.65	75.00	75.82	+ 2.38
+ Semantic Pruning	5	79.30	77.75	78.52	+ 2.70
+ Entry Recognition	5	74.33	72.57	73.44	- 2.82
+ Semantic Pruning	5	78.19	76.51	77.34	+ <b>3.90</b>
+ Syntactic Pruning	5	79.34	77.80	78.56	+ 2.30
+ Semantic Pruning	5	81.90	80.48	81.18	+ 2.62
+ Semantic Pruning	5	81.02	79.09	80.04	+ <b>3.78</b>

Table 3: Step-by-step results

selves WordNet entries. Our entry recognizer tends to do a good job since it managed to correctly identify all the relevant entries except in two cases (i.e. “take up” and “get rolling”), in which the dependency parse was incorrect. The drop in F1 for our automatic entry recognition was mainly due to incorrect selection of the correct entry of a set of alternative, overlapping entries.

Syntactic pruning did not prune the correct sense in most cases. In 16 cases (with gold entries), however, the correct sense was pruned. Five of these senses were pruned due to incorrect dependency parses, which led to incorrect frame identification. In two cases, the sense was not annotated with the recognized frame in WordNet, although it seemed adequate. In the remaining cases, a general frame from WordNet was incorrectly omitted. Improvements to WordNet’s frame annotations may thus make syntactic pruning even

more effective.

Semantic pruning also improves performance. Here the correct sense was pruned for 11 verbs, mainly due to the noisiness and incompleteness of our VOS repository. Without using gold entries, we found in total 237 semantic matches between possible verbs senses and possible object senses (200 with gold entries). We also found that our manual annotations in the VOS repository (see Sec. 6) did not affect our experiments.

The results show that syntactic and semantic pruning are beneficial for verb sense disambiguation, but also stress the necessity to improve existing resources. Ideally, each verb sense would be annotated with both the possible clause types or syntactic patterns in which it can occur as well as the possible senses of its objects. Annotations for subjects and adverbial arguments may also be beneficial.

## 8 Related Work

WSD is a classification task where for every word there is a set of possible senses given by some external resource (as a KB). Two types of methods can be distinguished in WSD. Supervised systems (Dang and Palmer, 2005; Dligach and Palmer, 2008; Chen and Palmer, 2009; Zhong and Ng, 2010) use a classifier to assign senses to words, mostly relying on manually annotated data for training. In principle, these systems suffer from low coverage since the training data is usually sparse. Some authors have tried to overcome this limitation by exploiting linked resources as training data (Shen et al., 2013; Cholakov et al., 2014).

The second WSD approach corresponds to the so-called KB methods (Agirre and Soroa, 2009; Ponzetto and Navigli, 2010; Miller et al., 2012; Agirre et al., 2014). They rely on a background KB (typically WordNet or extended versions (Navigli and Ponzetto, 2012)), where related senses appear close to each other. KB-based algorithms often differ in the way the KB is explored. It has been shown that a key point to enhance performance is the amount of semantic information in the KB (Ponzetto and Navigli, 2010; Miller et al., 2012). Our framework fits this line of work since it is also unsupervised and enriches the background knowledge in order to enhance performance of standard WSD algorithms. A comprehensive overview of WSD systems can be found in Navigli (2009) and Navigli (2012).

To bring WSD to real-world applications, the mapping between text and KB entries is a fundamental first step. It has been pointed that the existence of multi-word expressions imposes multiple challenges to text understanding tasks (Sag et al., 2002). The problem has been addressed by Arranz et al. (2005) and Finlayson and Kulkarni (2011). They find multi-word entries by matching word sequences allowing some morphological and POS variations according to predefined patterns. Our method differs in that we can recognize KB entries that appear discontinuously and in that we do not select the correct entry but generate a set of potential entries.

Linguists have noted the link between verb senses and the syntactic structure and argument types (Quirk et al., 1985; Levin, 1993; Hanks, 1996), and supervised WSD systems were developed to capture this relation (Dang and Palmer, 2005; Chen and Palmer, 2009; Dligach and

Palmer, 2008; Cholakov et al., 2014). In Dang and Palmer (2005) and Chen and Palmer (2009), it is shown that WSD tasks can be improved with features that capture the syntactic structure and information about verb arguments and their types. They use features as shallow named entity recognition and the hypernyms of the possible senses of the noun arguments. Dang and Palmer (2005) also included features extracted from PropBank (Palmer et al., 2005) from role labels and frames. Dligach and Palmer (2008) generated a corpus of verb and their arguments (both surface forms), which was used to incorporate a semantic feature to the supervised system.

In our work, we also incorporate syntactic and semantic information. Instead of learning the relation between the verb senses and the syntactic structure, however we incorporate it explicitly using the WordNet frames, which provide information about which verb sense should be considered for a given syntactic pattern. We also incorporate explicitly the semantic relation between each verb sense and its arguments using our VOS repository.

Different resources of semantic arguments for automatic text understanding tasks have been constructed (Baker et al., 1998; Palmer et al., 2005; Kipper et al., 2008; Gurevych et al., 2012; Nakashole et al., 2012; Flati and Navigli, 2013). In (Baker et al., 1998; Palmer et al., 2005; Kipper et al., 2008; Gurevych et al., 2012), the classification of verbs and arguments is focused toward semantic or thematic roles. Nakashole et al. (2012) uses semantic types to construct a taxonomy of binary relations and Flati and Navigli (2013) collected semantic arguments for given textual expressions. For instance, given the verb “break”, they extract a pattern “break *<body part-1>*”. In contrast to existing resources, our VOS repository disambiguates both the verb sense and the senses of its arguments.

## 9 Conclusion

We presented Werdy, a framework for word-sense recognition and disambiguation with a particular focus on verbs and verbal phrases. Our results indicate that incorporating syntactic and semantic constraints improves the performance of verb sense disambiguation methods. This stresses the necessity of extending and improving the available syntactic and semantic resources, such as WordNet or our VOS repository.

## References

- Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of EACL*, pages 33–41.
- Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.
- Victoria Arranz, Jordi Atserias, and Mauro Castillo. 2005. Multiwords and word sense disambiguation. In *Computational Linguistics and Intelligent Text Processing*, volume 3406 of *Lecture Notes in Computer Science*, pages 250–262.
- B. T. Sue Atkins and Michael Rundell. 2008. *The Oxford Guide to Practical Lexicography*. Oxford University Press.
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of ACL*, pages 86–90.
- Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of IJCAI*, pages 805–810.
- Yee Seng Chan, Hwee Tou Ng, and Zhi Zhong. 2007. Nus-pt: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proceedings of SemEval*, pages 253–256.
- Jinying Chen and Martha Palmer. 2009. Improving english verb sense disambiguation performance with linguistically motivated features and clear sense distinction boundaries. *Language Resources and Evaluation*, 43(2):181–208.
- Kostadin Cholakov, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Automated verb sense labelling based on linked lexical resources. In *Proceedings of EACL*, pages 68–77.
- Hoa Trang Dang and Martha Palmer. 2005. The role of semantic roles in disambiguating verb senses. In *Proceedings of ACL*, pages 42–49.
- Luciano Del Corro and Rainer Gemulla. 2013. Clause: clause-based open information extraction. In *Proceedings of WWW*, pages 355–366.
- Dmitriy Dligach and Martha Palmer. 2008. Improving verb sense disambiguation with automatically retrieved semantic knowledge. In *Proceedings of ICSC*, pages 182–189.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Mark Alan Finlayson and Nidhi Kulkarni. 2011. Detecting multi-word expressions improves word sense disambiguation. In *Proceedings of MWE*, pages 20–24.
- Mark Alan Finlayson. 2014. Java libraries for accessing the princeton wordnet: Comparison and evaluation. In *Proceedings of GWC*.
- Tiziano Flati and Roberto Navigli. 2013. Spred: Large-scale harvesting of semantic predicates. In *Proceedings of ACL*, pages 1222–1232.
- Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M. Meyer, and Christian Wirth. 2012. Uby - a large-scale unified lexical-semantic resource based on lmf. In *Proceedings of EACL*, pages 580–590.
- Patrick Hanks. 1996. Contextual dependency and lexical sets. *International Journal of Corpus Linguistics*, 1(1):75–98.
- Adam Kilgarriff and Joseph Rosenzweig. 2000. Framework and results for english senseval. *Computers and the Humanities*, 34(1-2):15–48.
- Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation*, 42(1):21–40.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*, pages 423–430.
- Shari Landes, Claudia Leacock, and Randee I. Tengi. 1998. *Building Semantic Concordances*. MIT Press.
- Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC*, pages 24–26.
- Beth Levin. 1993. *English Verb Classes and Alterations: A Preliminary Investigation*. University of Chicago Press.
- Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proceedings of COLING*, pages 1781–1796.
- Tristan Miller, Nicolai Erbs, Hans-Peter Zorn, Torsten Zesch, and Iryna Gurevych. 2013. Dkpro wsd: A generalized uima-based framework for word sense disambiguation. In *Proceedings of ACL: System Demonstrations*, pages 37–42.
- Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of EMNLP*, pages 1135–1145.
- Roberto Navigli and Mirella Lapata. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.

- Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193(0):217–250.
- Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of SemEval*, pages 30–35.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69.
- Roberto Navigli. 2012. A quick tour of word sense disambiguation, induction and related approaches. In *Proceedings of SOFSEM*, pages 115–129.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Ted Pedersen, Siddharth Patwardhan, and Jason Mitchell. 2004. Wordnet::similarity: Measuring the relatedness of concepts. In *Proceedings of HLT-NAACL: Demonstration Papers*, pages 38–41.
- Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of ACL*, pages 1522–1531.
- Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.
- Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Proceedings of CICLing*, pages 1–15.
- Hui Shen, Razvan Bunescu, and Rada Mihalcea. 2013. Coarse to fine grained sense disambiguation in wikipedia. In *Proceedings of \*SEM*, pages 22–31.
- Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of ACL: System Demonstrations*, pages 78–83.