Toward A Semantic Concordancer

Adam Pease
Articulate Software
San Jose, USA
apease@articulatesoftware.com

Andrew K. F. Cheung
Hong Kong Polytechnic University
Hong Kong SAR, China
andrew.cheung@polyu.edu.hk

Abstract

Concordancers are an accepted and valuable part of the tool set of linguists and lexicographers. They allow the user to see the context of use of a word or phrase in a corpus. A large enough corpus, such as the Corpus Of Contemporary American English, provides the data needed to enumerate all common uses or meanings.

One challenge is that there may be too many results for short search phrases or common words when only a specific context is desired. However, finding meaningful groupings of usage may be impractical if it entails enumerating long lists of possible values, such as city names. If a tool existed that could create some semantic abstractions, it would free the lexicographer from the need to resort to customized development of analysis software.

To address this need, we have developed a Semantic Concordancer that uses dependency parsing and the Suggested Upper Merged Ontology (SUMO) to support linguistic analysis at a level of semantic abstraction above the original textual elements. We show how this facility can be employed to analyze the use of English prepositions by non-native speakers.

We briefly introduce condordancers and then describe the corpora on which we applied this work. Next we provide a detailed description of the NLP pipeline followed by how this captures detailed semantics. We show how the semantics can be used to analyze errors in the use of English prepositions by non-native speakers of English. Then we provide a description of a tool that allows users to build semantic search specifications from a set of English examples and how those results can be employed to build rules that translate sentences into logical forms. Finally, we summarize our conclusions and mention future work.

1 Introduction

Concordancers[1] enable the linguist to see the context of use of words or phrases. This is valuable in understanding how a word can have different senses, or in finding rules or exceptions for collocations. One issue for the linguist using such tools is that many linguistic constructions are *patterns* or types, rather than literal collections of words. We "take a pill" but "eat a muffin", we "play music" but "draw a picture", "fly a plane" but "drive a car" or "pilot a boat". For each of the nouns, a class or group determines the verb (such as "medicine", "2-D art" or "aircraft"), but enumerating those possibilities is cumbersome. A computational linguist could develop customized analysis software, but no general purpose tool fit for this task appears to exist. We have developed software that allows the linguist to specify dependency relations and semantic types, based on a formal ontology, that can alleviate the need to enumerate large numbers of alternative strings of search terms with a conventional concordancer.

2 Corpora

To motivate development of this software we have two use cases. The first case is in analysis of corpora for classes of errors in usage that are common for non-native speakers of English. We chose to look at a small corpus of translated speech and analyze it for these classes of errors. In this way, we can provide specific feedback to translators on

[1] such as http://www.antlab.sci.waseda.ac.jp/antconc_index.html and https://www.lextutor.ca/conc/eng/

what problems to avoid in the future. To augment this work, we also examined a larger and broader corpus of non-native English usage, in order to help validate the utility of the tool on a corpus that has more, and more obvious, usage errors. We begin with a corpus of legal judgments translated from Chinese into English.

Judgments translated from Chinese into English are essential to the rule of law in Hong Kong. Hong Kong is the only common law jurisdiction where Chinese and English languages are used alongside each other in the judicial system (Cheng and He, 2016). Judgments form an essential part of common law. Because the majority of the population is Chinese speaking, court cases are sometime heard in Chinese. Judgments in these Chinese cases are written in Chinese. Judgments of cases with jurisprudence value are translated into English. These translated judgments may be used in the future by legal professionals who are not necessarily familiar with the Chinese language. Translated English judgments were downloaded from the Hong Kong Judiciary website[2] to build the Hong Kong translated English judgments corpus.

Non-native speakers can find it challenging to use English prepositions properly. Compared to English, Chinese is a verb heavy language. The Chinese language has significantly fewer prepositions than the English language does. Unlike English, Chinese sentences without prepositions are grammatically correct and comprehensible (Shih, 2012). Chinese speakers, even with good English language abilities, may not be as sensitive to the use of prepositions when using the English language. Therefore, one of the challenges facing Chinese speakers when translating into English is the accurate use of prepositions.

After removing titles, headings and other incomplete sentences in the legal corpus, we arrived at 8818 sentences in suitable for further processing by our semantic concordancer.

To broaden our study, we also examined the Cambridge Learner Corpus (CLC)[3] (Yannakoudakis et al., 2011), which has a greater number of English usage errors and is roughly twice the size of our legal corpus, at 16068 lines of text, also ignoring titles and headings.

Our second use case is in validating linguistic patterns and creating rules to translate language to logical forms, for which we employ two large corpora of native English writing. These are the Corpus Of Contemporary American English (COCA) (Davies, 2008) and 2722 articles from Wikipedia converted to plain text[4].

3 NLP Pipeline

Our work relies upon the Stanford CoreNLP (Duchi et al., 2011) pipeline, which is free and open source, and either the top performing system or at least state of art on each element of its pipeline. The system is structered as a series of *annotations* on tokens. Each annotator builds up annotations on the textual input.

To illustrate the pipeline, let's take a particular example.

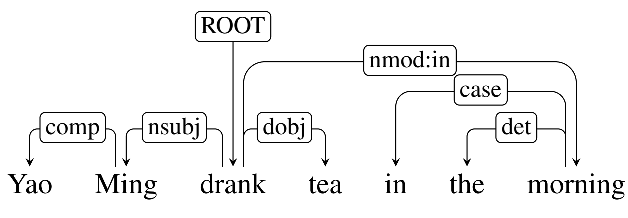
(1) Yao Ming drank tea in the morning.

The Stanford Named Entity Recognizer (Finkel et al., 2005) identifies linguistic references to things like names and dates. It results in the following markings of our example (where "O" is a tag for "other", meaning not a named entity)

(2)	PERSON	PERSON	O	O	O	O	TIME
	Yao	Ming	drank	tea	in	the	morning

We have added a multi-word phrase recognizer to the CoreNLP pipeline that uses WordNet and SUMO as dictionaries. Matching multi-word elements are reduced to a single token, so "Yao Ming" or "July 23rd" will become a single token with a class membership in SUMO (`Human` or `Day` respectively here).

Dependency parsing (Chen and Manning, 2014) abstracts parse trees into a set of linguistic relations that are as independent of language as possible. We have the following dependency graph for example 1:



Note that dependencies as a data structure can also be represented as just a list of triples.

[2] http://www.judiciary.hk/en/index/

[3] https://www.ilexir.co.uk/datasets/index.html

[4] http://www.evanjones.ca/software/wikipedia2text.html

```
root(ROOT-0,drank-2)
compound(Ming-2,Yao-1)
nsubj(drank-3,Ming-2)
dobj(drank-3,tea-4)
case(morning-7,in-5)
det(morning-7,the-6)
nmod:in(drank-3,morning-7)
```

CoreNLP lacks a module for determining word senses so we have utilized our existing system from (Pease and Li, 2010). This process normally addresses just nouns, verbs, adjectives and adverbs. Determining named entities is done in the NER system described earlier. WSD annotations are shown as example 3, and definitions for some different senses of "tea" are shown in table 1.

(3) Yao Ming drank tea in the morning
. . 201170052 107933274 . . 115165289

These IDs are for WordNet 3.0 (Fellbaum, 1998) (with the part of speech number prepended) and they have been manually linked to the Suggested Upper Merged Ontology (SUMO)[5] (Niles and Pease, 2001; Pease, 2011). Since the original mapping effort in 2002, tens of thousands of synsets have been remapped to more specific SUMO terms as they have been defined. In particular, several thousand have been remapped in 2017 alone. The current statistics for the mappings are shown in Table 2. Note that a small number of adjectives and adverbs have not been mapped.

Instance mappings are from a SUMO term to a particular instance synset in WordNet, such as SUMO's `Battle` mapping to WordNet's "Battle of Britain". *Equivalence* mappings are close but informal equivalences, such as the mapping between SUMO's `Cloud` and WordNet's synset 109247410 "a visible mass of water or ice particles suspended at a considerable altitude." *Subsuming* mappings are between specific WordNet synsets and more general SUMO terms, such as "Meniere's_disease" and SUMO's `DiseaseOrSyndrome`. Of note is that recently, with the growth of SUMO in several domains, we increasingly have need for what we might term a "subsumed-by" relation, where a SUMO term is more specific than any available WordNet synset, as is the case with the new ontologies of Law and Weather. This relation is likely to appear in a future release of the mappings. We also augment the WordNet lexicon with lexical entries provided in the ontology for each new term, such as the string "mono crystalline" being associated with the recently-added SUMO term `MonoCrystalline`.

To perform word sense disambiguation, we rely on WordNet SemCor (Landes et al., 1998), a corpus of manually-marked word senses, indexed to the WordNet semantic lexicon, and annotated on the Brown Corpus of English (Kucera and Francis, 1967). For each word sense, we create a table counting the frequency of co-occurring words in the corpus. We use a frequency threshold so that low-frequency senses that have little co-occurrence data aren't influenced by random small amounts of data. One criticism of WordNet has been that it makes some overly fine distinctions among word senses (Snow et al., 2007). We use the SUMO-WordNet mappings to collapse senses that map to the same term in the ontology. Note however that this grouping is much more fine grained than the coarse-grained aggregation to categories done in SemEval-17 on OntoNotes (Pradhan et al., 2007b), so that fewer (if any) meaningful distinctions in sense are lost. This approach has the added benefit of increasing the statistical significance of some of the merged cooccurrence relationships. This approach however does not perform as well as some recent effort in WSD that employ machine learning, such as (Zhong and Ng, 2010). When tested on the OntoNotes corpus (Pradhan et al., 2007a) we achieve roughly 66% accuracy, which approaches the score (stated at 72% in (Brown et al., 2010)) for inter-annotator agreement on fine grained senses. Since we cannot assume a particular domain, accuracies are likely to be lower than the best results of other reported studies (Zhong et al., 2008). However, it is likely that more training data from a wider set of corpora[6] will help improve performance.

We augment Stanford dependency parses with SUMO terms. Continuing the example above, we add the triples

```
sumo(Drinking,drank-3)
sumo(Morning,morning-7)
sumo(Tea,tea-4)
```

While SUMO does have a taxonomy, it also has definitions in a higher order logic that explain, in a computable way, the meaning of each term. So,

[5] http://www.ontologyportal.org

[6] https://github.com/getalp/LREC2018-Vialetal

sense key	words	definition
107575510	tea, teatime	a light midafternoon meal
107933274	tea	a beverage made by steeping tea leaves in water
107932841	tea, tea_leaf	dried leaves of the tea shrub

Table 1: Word senses (definitions and word lists shortened from WordNet)

	instance	equivalence	subsuming
noun	9,570	6,505	67,914
verb	0	971	13,204
adjective	730	596	14,832
adverb	57	119	3,222
total	10,357	8191	99,172

Table 2: SUMO-WordNet mapping statistics (117,720 total synsets mapped)

for the example of `Drinking` we have logical axioms such as

```
(=>
    (attribute ?A Thirsty)
    (desires ?A
        (exists (?D)
            (and
                (instance ?D Drinking)
                (agent ?D ?A)))))
```

that states that being `Thirsty` implies a desire to drink something. Axioms such as this are more specific and detailed than entailment links and can enable further logical reasoning.

We have linked the Stanford 7-class NER model to SUMO types, which allows us to assert

```
sumo(Human,Yao_Ming-1)
```

from the NER output shown in example 2.

We also employ Stanford's SUTime (McClosky and Manning, 2012) to recognize temporal expressions. If we have the slightly modified example

(4) Yao Ming drank tea in July.

we would add the clauses.

```
month(time-1,July)
time(drank-3,time-1)
```

Although the current semantic concordancer system does not employ logical deduction, the information captured would allow us to use SUMO's temporal axioms and its associated E Theorem Prover (Pease and Schulz, 2014) to do simple temporal reasoning, and further expand the possibilities of searching for semantic patterns to include relative periods like "before June" or "during 2016" and return sentences that meet those constrants rather than a literal pattern of words.

4 Semantic Concordance

Concordancers are very useful for checking intuitions with respect to language usage. Searching on a word or phrase provides samples of usage in context. But not all language patterns are strict phrases. Idioms can have insertions (Minugh, 2007), such as "drop in the bucket" being modified to "drop in the proverbial bucket" or "drop in the fiscal bucket" but not "He put a drop of water in the bucket". Being able to search a dependency parse for a grammatical pattern rather than a literal string or even a string with wildcards may be a useful tool.

Some patterns of usage are selected with respect to the types of participants in a phrase, rather than particular words. These can be quite specific. For example, if a linguist wants to examine usage of the preposition "in" in its physical, rather than temporal sense, an exhaustive number of searches would be required to enumerate physical words and phrases and temporal words or phrases. However, given that we have dependency parse forms and SUMO terms we can search for patterns such as:

```
  nmod:in(?X,?Y), sumo(?C,?Y),
isSubclass(?C,TimePosition)
  nmod:in(?X,?Y), sumo(?C,?Y),
isSubclass(?C,Object)
```

To carry on with example 1, note how the first pattern involving `TimePosition` above matches with the clauses of the example, and the variables are bound to `?X=drank-3`, `?Y=morning-7` and `?C=Morning`.

```
root(ROOT-0,drank-3)
det(morning-7, the-6)
nmod:in(drank-3,morning-7)
sumo(Human,Yao_Ming-1)
sumo(Drinking,drank-3)
sumo(Morning,morning-7)
names(Yao_Ming-1,"Yao")
dobj(drank-3,tea-4)
case(morning-7,in-5)
sumo(Tea,tea-4)
names(Yao_Ming-1,"Ming")
nsubj(drank-3,Yao_Ming-1)
```

While WordNet noun synsets could be used to capture common classes of words, SUMO provides extra utility when searching for groups of verbs. For example, one "looks for" or "searches for" something in order to find it and some language learners may omit the preposition. In each case there is a mapping to SUMO's `Searching`, but no common hypernym for those WN 3.0 senses (201315613 and 202153709, respectively).

Because WSD and dependency parsing are not always correct, it is necessary to review results rather than simply tabulating them. Also, language is flexible, and what constitutes "correct" usage is more like correspondence to a preponderance of use than a strict rule in many cases.

5 Preposition Errors

We looked for common errors in preposition usage[7] in our corpora of non-native English. The first error type that was searched for was the use of prepositions with times of day (see example 5), where "night" is an exception.

(5) ... in the morning ...
* ... at the morning ...
... in the evening ...
* ... at the evening ...
... at night ...
* ... in night ...

We can state the (ungrammatical) dependency pattern

```
nmod:at(?X,?Y), sumo(?C,?Y),
isSubclass(?C,TimeInterval)
```

One sentence found in the corpus was example 6,

(6) "We usually have lessons at the morning, till afternoon."

[7] `http://blog.oxforddictionaries.com/2017/01/preposition-mistakes-for-english-learners/`

This sentence has the augmented dependency parse of

```
root(ROOT-0, have-3)
nsubj(have-3, We-1)
advmod(have-3, usually-2)
dobj(have-3, lessons-4)
case(morning-7, at-5)
det(morning-7, the-6)
nmod:at(lessons-4, morning-7)
case(afternoon-10, till-9)
nmod:till(have-3, afternoon-10)
sumo(SubjectiveAssessmentAttribute,
usually-2)
sumo(EducationalProcess,lessons-4)
sumo(Morning,morning-7)
sumo(Afternoon,afternoon-10)
```

Other examples of linguistic errors in the corpus found by matching dependency patterns are

(7) * I've been working here since five years.
* If Tang Dan-dan was also manipulated as was the applicant, she should have arrived at Hong Kong as scheduled.

6 Query Composition

One of the challenges in using this tool is that it requires some knowledge of dependency parsing and SUMO. To address this, we have created a component that find the common structure of several sentences and returns a dependency parse for that common structure. That specification can then be used to search for other sentences that match the pattern. In this way, the linguist simply has to prepare several sentences that illustrate a common construction and let the system do the work to state the commonality in a formal language.

Take for example the following two sentences

(8) John kicks the cart.

(9) Susan pushes the wagon.

which produce the following respective augmented dependency parses -

```
root(ROOT-0,kicks-2)
det(cart-4,the-3)
names(John-1,"John")
sumo(Wagon,cart-4)
sumo(Kicking,kicks-2)
nsubj(kicks-2,John-1)
dobj(kicks-2,cart-4)
attribute(John-1,Male)
sumo(Human,John-1)
```

```
root(ROOT-0,pushes-2)
det(wagon-4,the-3)
names(Susan-1,"Susan")
attribute(Susan-1,Female)
sumo(Pushing,pushes-2)
sumo(Human,Susan-1)
dobj(pushes-2,wagon-4)
nsubj(pushes-2,Susan-1)
sumo(Wagon,wagon-4)
```

We can then produce their common, unified abstraction as follows, in which labels with question marks denote variables -

```
root(ROOT-0,?B)
det(?D,?C)
names(?A,?E)
attribute(?A,SexAttribute)
sumo(Motion,?B)
sumo(Human,?A)
dobj(?B,?D)
nsubj(?B,?A)
sumo(Wagon,?D)
```

Note that the expression can be verified to unify with the original dependency parses, using the following substitutions for sentence 8 as an example.

```
?A=John-1
?B=kicks-2
?C=the-3
?D=cart-4
```

A linguist who does not have the facility to write dependency parses or use SUMO can simply use the resulting expression as a "black box" search input to the concordancer. A future version of the system could even have an option to hide it entirely, thereby performing a form of semantic search.

7 Semantic Rewriting

The Semantic Concordancer is an intermediate result from efforts to translate language into logic. We are extending prior work on the Controlled English to Logic Translation (Pease and Li, 2010) to use modern parsing techniques with Stanford's CoreNLP instead of a restricted English grammar.

When the semantics of sentences are fully captured it opens up opportunities for deductive reasoning that goes beyond simple retrieval of previous sentences. It also creates the possibility to vet utterances for contradictions with known facts about the world, thereby allowing a system to exclude faulty parses based on world knowledge.

For example, the simple sentence 8 above becomes the following first-order logic sentence with SUMO terms -

```
(exists (?John-1 ?cart-4 ?kicks-2)
  (and
    (agent ?kicks-2 ?John-1)
    (attribute ?John-1 Male)
    (names ?John-1 "John")
    (patient ?kicks-2 ?cart-4)
    (instance ?cart-4 Wagon)
    (instance ?kicks-2 Kicking)
    (instance ?John-1 Human)) )
```

The process of accomplishing this is what we call Semantic Rewriting, and is based on previous efforts called Transfer Semantics or Packed Rewriting (Crouch, 2005; Crouch and King, 2006). It involves the iterative application of production rules to dependency parses. In the case of sentence 8 this involves execution of just two rules (along with a simple mechanical listing of the types of terms with `instance` and generation of the name of "John" as a male human from a common name database) -

```
dobj(?E,?Y) ==> (patient(?E,?Y)).
  line 1041 : {?E=kicks-2, ?Y=cart-4}

nsubj(?E,?X), sumo(?A,?E),
  isSubclass(?A,Process), sumo(?C,?X),
  isSubclass(?C,Agent) ==> (agent(?E,?X)).
  line 1063 :
  {?X=John-1, ?A=Kicking, ?C=Human,
   ?E=kicks-2}
```

The first rule is a general default that if we have no more specific pattern, the direct object in a sentence becomes the "patient" in a SUMO expression. The second rule is more interesting. It states that if the grammatical subject of a `Process` is an `Agent` (rather than some inanimate object) then we generate a SUMO `agent` relationship between the entity and the process.

While creating a few simple rules of this sort is easy, as the rule set grows and the remaining rules become more complex, authoring them through introspection become impractical. The Query Composition tool described above provides a principled way to create patterns by example, which form the left hand side of a Semantic Rewriting rule. The Semantic Concordancer then becomes useful as a way to validate the prevalence of a particular pattern of language use in a large corpus.

8 Conclusions and Future Work

The software is available open source at `https://github.com/ontologyportal` and has been used on a practical pilot project in analysis of non-native English. We expect to apply it further to more systematic studies in this area as well as others. The implementation is in Java, using the H2 database[8]. All the words in each sentence and terms in dependency parses are indexed, so all semantic processing occurs at the time the database is built, rather than when a query is run. After sentences and dependencies matching a bag of terms are returned, a simple unification algorithm attempts to match the dependency parse literals with the dependency parse query, similar to a Prolog-style unification algorithm (Baader and Snyder, 2001). This enables the system to scale well to the requirements of modern large corpora.

We are employing the Semantic Concordancer and its associated Query Composition tool to create and validate semantic rules that translate language into logical expressions.

The system will be available by the time of GWC2018 on a server at `https://nlp.ontologyportal.org:8443/sigmanlp/semconcor.jsp`.

References

Baader, F. and Snyder, W. (2001). Unification theory. In *Handbook of Automated Reasoning (in 2 volumes)*, pages 445–532.

Brown, S. W., Rood, T., and Palmer, M. (2010). Number or nuance: Which factors restrict reliable word sense annotation? In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*.

Chen, D. and Manning, C. D. (2014). A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of EMNLP 2014*.

Cheng, L. and He, L. (2016). Revisiting judgment translation in hong kong. *Semiotica*, (209):59–75.

Crouch, R. (2005). Packed rewriting for mapping semantics to KR. In *Proc. 6 th Int. Workshop on Computational Semantics*, pages 103–114, Tilburg.

Crouch, R. and King, T. H. (2006). Semantics via f-structure rewriting. In *Proceedings of LFG06*, pages 145–165.

Davies, M. (2008). The corpus of contemporary american english (coca): 520 million words, 1990-present. available online at https://corpus.byu.edu/coca/.

[8]`www.h2database.com/`

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370.

Kucera and Francis, W. N. (1967). *Computational Analysis of Present-Day American English*. Providence: Brown University Press.

Landes, S., Leacock, C., and Tengi, R. (1998). Building semantic concordances. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, Language, Speech, and Communication. MIT Press, Cambridge (Mass.).

McClosky, D. and Manning, C. D. (2012). Learning constraints for consistent timeline extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 873–882, Stroudsburg, PA, USA. Association for Computational Linguistics.

Minugh, D. C. (2007). The filling in the sandwich: internal modification of idioms. *Language and Computers*, 62(1):205–224.

Niles, I. and Pease, A. (2001). Toward a Standard Upper Ontology. In Welty, C. and Smith, B., editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 2–9.

Pease, A. (2011). *Ontology: A Practical Guide*. Articulate Software Press, Angwin, CA.

Pease, A. and Li, J. (2010). Controlled English to Logic Translation. In Poli, R., Healy, M., and Kameas, A., editors, *Theory and Applications of Ontology*. Springer.

Pease, A. and Schulz, S. (2014). Knowledge engineering for large ontologies with sigma kee 3.0. In Demri, S., Kapur, D., and Weidenbach, C., editors, *Automated Reasoning: 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 19-22, 2014. Proceedings*, pages 519–525. Springer International Publishing.

Pradhan, S. S., Hovy, E. H., Marcus, M. P., Palmer, M., Ramshaw, L. A., and Weischedel, R. M. (2007a). Ontonotes: a unified relational semantic representation. *Int. J. Semantic Computing*, 1(4):405–419.

Pradhan, S. S., Loper, E., Dligach, D., and Palmer, M. (2007b). Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 87–92, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shih, C. L. (2012). A corpus-aided study of shifts in english-to-chinese translation of prepositions. *International Journal of English Linguistics*, 2(6).

Snow, R., Prakash, S., Jurafsky, D., and Ng, A. Y. (2007). Learning to merge word senses. In *EMNLP-CoNLL*, pages 1005–1014. ACL.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). A New Dataset and Method for Automatically Grading ESOL Texts. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Zhong, Z. and Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 78–83, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhong, Z., Ng, H. T., and Chan, Y. S. (2008). Word sense disambiguation using ontonotes: An empirical study. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1002–1010.