WordNet-Based Lexical Simplification of a Document

S. Rebecca Thomas
Bard College
Annandale-on-Hudson, NY
USA 12504
thomas@bard.edu

Sven Anderson
Bard College
Annandale-on-Hudson, NY
USA 12504
sanderso@bard.edu

Abstract

We explore algorithms for the automatic generation of a limited-size lexicon from a document, such that the lexicon covers as much as possible of the semantic space of the original document, as specifically as possible. We evaluate six related algorithms that automatically derive limited-size vocabularies from Wikipedia articles, focusing on nouns and verbs. The proposed algorithms combine Personalized Page Rank (Agirre and Soroa, 2009) and principles of information maximization, beginning with a user-supplied document and constructing a customized small vocabulary using WordNet. The best-performing algorithm relies on word-sense disambiguation with sentence-level context information at the earliest stage of analysis, indicating that this computationally costly task is nonetheless valuable.

1 Introduction

This report explores algorithms for the automatic generation of a limited size lexicon from a document, such that the lexicon covers as much as possible of the semantic space of the original document, as specifically as possible. While lexical simplification has typically worked at the sentence level, it is our belief that the document as a whole must be taken into account, and that creating a simpler vocabulary for the document and then applying that vocabulary to the sentences will give more appropriate results. This paper presents some results on the problem of developing a document-level vocabulary.

Tasks for which it would be desirable to generate a smaller, and ideally simpler, vocabulary include full-document text summarization and text paraphrase. In addition, simplification of a vocabulary can be extended to icon sets used by mobile devices or augmentative communication devices. In particular, one type of Augmented and Assistive Communication (AAC) system includes a touchscreen, from which users select icons that may, alone or in combination, represent specific words or phrases (Baker, 1982).[1] An AAC icon set represents a set of concepts customized by a human expert to a particular user. There is tension between ensuring that the collection of icons – the vocabulary – is large enough to be sufficiently expressive and ensuring that it is small enough to allow for efficient navigation and maximal communication speed, a major issue with AAC systems (Trnka et al., 2009). One might design the icon set beginning with a text corpus representing typical utterances of the particular user – perhaps from a log file kept by a communication device – and generate an icon set with more, and more specific, icons for topics of frequent communication. Likewise, for any device with a touchscreen interface, reducing the "vocabulary" of touchable icons without compromising expressivity allows for better usage of limited screen area.

In the next section we outline previous relevant work in simplification. The third section describes six methods for generating a reduced

[1]Users with appropriate levels of literacy and motor control may be able to type on an alphabetic keyboard; these are not the users we are primarily concerned with in this work, but see (Wandmacher et al., 2008; Trnka et al., 2009).

Proceedings of KONVENS 2012 (Main track: oral presentations), Vienna, September 19, 2012

lexicon from a document-derived vocabulary and describes two measures for evaluating the quality of the resulting lexicons. The fourth section discusses results obtained with English Wikipedia (EW) articles. The paper concludes with discussion and interpretation of our results.

2 Background

Text simplification is usually implemented in two stages: first syntactic analysis and simplification and then lexical simplification, which reduces the number of difficult words and expressions. The earliest approach to lexical simplification replaces words by simpler synonyms. Synonym difficulty is estimated using the frequency and length of each word: more difficult words usually have lower frequency of occurrence and a greater number of syllables. For example, (Carroll et al., 1998) queries WordNet for synonyms (synsets) and selects the most frequently occurring synonym in a synset as determined by frequency (Kucera and Francis, 1967). The authors wanted to avoid deep semantic analysis; thus, no word-sense disambiguation was performed. Lal (2002) extended this approach by including the syllable count as part of a word's difficulty metric. More recently, lexical and syntactic simplification have been simultaneously implemented via statistical machine translation (Coster and Kauchak, 2011) and English Wikipedia edit histories (Yatskar et al., 2010). These studies did not directly evaluate the success of lexical simplification.

DeBelder et al. (2010) add a form of word-sense disambiguation to lexical simplification. The authors use the latent variables from a Bayesian network language model to suggest alternate words in a text. An advantage of this approach is that it does not necessarily rely on specialized semantic databases like WordNet, nor does it rely on psychological data to estimate word difficulty. DeBelder and Moens (2010) use this approach to simplify the syntax and lexicon of text for children. They show an improvement of about 12% in accuracy, as assessed by human judges, over the baseline model that simply chooses the most frequent synonym.

Lexical simplification within a sentence context was a SemEval-2012 task (Specia et al., 2012). The best of five submitted systems employed word frequency as well as context-dependent information. The authors conclude that research is needed that evaluates the role of context in the simplification of all words in a context. The present paper extends preliminary results in (Anderson et al., 2011), describing derivation of a small, simplified vocabulary in which the context is the entire document: lexical choice decisions are made simultaneously and for an entire document, not sentence by sentence.

3 Reducing Vocabulary Size

3.1 Algorithms

We developed and tested six methods for generating the reduced lexicon. Given a starting document in standard English, we tag the text using the Stanford Part of Speech Tagger (Toutanova and Manning, 2000), extract all and only the occurrences of the part of speech we are interested in, and reduce each word occurrence to its base uninflected form using WordNet's morphstr method. [2] We explored three levels of word-sense disambiguation: none, weak, and strong disambiguation. This step produced a base set from which we generate a reduced lexicon using frequency-based selection, weights generated by disambiguation, or a greedy algorithm described below. We discuss the disambiguation step and then the vocabulary reduction step, concluding with an overview of the resulting six algorithms.

Disambiguation Step

Both strong and weak word-sense disambiguation employed the approach of (Agirre and Soroa, 2009).[3] The PageRank algorithm underlying PPR permits synsets to vote for one another depending on WordNet's graph structure and the weight of each synset. This voting process is iterated until it converges to a stable solution that ranks vertices in a graph based on their relative importance. In weak disambiguation, each word appearing in the starting document is weighted proportional to the count of that word. All other synsets are initialized with weight zero, and PPR is performed once for the entire document. Strong disambiguation processes words one at a time by weighting

[2] We limit our focus to nouns and verbs, working with each part of speech independently.

[3] http://ixa2.si.ehu.es/ukb/

its neighboring words in the original text. Agirre and Soroa obtained superior disambiguation performance with this method, which they call *w2w*.

Lexicon Reduction Step

The lexical reduction step begins with the set of word senses from the first step and reduces these to a lexicon of any desired size. The two simplest algorithms merely select those N words with the highest frequency or those with the highest PPR weight after convergence.

We contrast simple reduction with a "greedy" information approach. The greedy algorithm constructs a subtree of WordNet containing all the words of a base set constructed by the first step. The WordNet graph for nouns is a tree with the word *entity* as it root. The graph for verbs is not a tree, lacking a common unique ancestor node; there are at least 15 different head nodes, and hundreds of verbs with no unique ancestor (Richens, 2008). We force a tree structure for the verb graph by adding a new node, which is made the parent of all the root nodes in the original verb forest.

We add to each tree node a count: for leaves, this is the number of occurrences of that word sense in the original document; for internal nodes, it is the sum of its own occurrences and its children's counts.

The objective of lexical reduction is to derive an optimal set of hypernyms, H, for the document. Initially H contains only the root node. The algorithm works its way down the constructed subtree, at each step evaluating every child of every node already in H, and greedily adding the child that maximizes information gain to the growing lexicon; see Figure 1. Note that the sequence of choices of locally optimal children may not lead to a globally optimal solution.

The information gained when a child is added to the hypernym set is computed via

$$-p * \left(\frac{c}{p}\log\left(\frac{c}{p}\right) + \left(\frac{p-c}{p}\right)\log\left(\frac{p-c}{p}\right)\right)$$

where c is the number of word occurrences covered by the child synset and p is the number of word occurrences covered by its parent. At each iteration, this guarantees that the node that maximizes information gain, given prior choices, is added to H; the number of word occurrences it covers is subtracted from its parent's count, as the newly added node is now the covering synset for those occurrences. If the parent synset no longer covers any word occurrences – that is, if its count reaches zero – then it is removed from H. Iterations continue until H reaches its target size.

Algorithms

Combining disambiguation and lexical reduction steps, we implemented and evaluated six of the resulting algorithms to create lexicons of prespecified size N.

Frequency-based (Baseline) As a baseline, we follow the approach of (Carroll et al., 1998) and employ word frequency as a surrogate measure of simplicity. After disambiguating words using strong disambiguation (w2w), we then select as our lexicon those N words which have the greatest combined frequency as measured by the product of document frequency and frequency of occurrence in the Web1T corpus (Brants and Franz, 2006). Only unigram and bigram frequencies were employed. Given any word in the original vocabulary, we can traverse up the tree from it until we reach the first node that is a member of the lexicon; we say this is the lexicon element that *covers* the word. Note that a lexicon H will not necessarily cover every word in the initial document.

No Disambiguation (Greedy) No disambiguation is used. Each word's number of occurrences is credited to every one of its senses, so the base set consists of all senses of every word (that is the proper part of speech) in the document. Lexical reduction employs the Greedy algorithm.

Personalized PageRank, Top-N (PPR-N) For each word appearing in the starting document, we find the corresponding synset(s) in WordNet and assign them weights proportional to the count of occurrences of that word. All other synsets are initialized with weight zero. After convergence of PPR, the N highest-weight synsets in the result comprise the reduced lexicon H. Again, H may not cover every word in the initial document.

Estimated Proportions of Word Sense Occurrences (PPR-WSD-G) Full PPR-W2W word sense disambiguation is time-consuming, requir-

```
Initialize hypernym set H to contain the root of the tree: {entity}
While H has not reached its final size,
    for each child c of each element p in H,
        compute information gained if c is added to H
    let x be the child that maximizes information gain; insert x into H
    subtract x’s count from its parent’s count
    if x’s parent no longer covers any vocabulary, remove x’s parent from H
```

Figure 1: Pseudocode for greedy algorithm

ing hours for longer articles. In the PPR-WSD-G approach, rather than performing full disambiguation of each word occurrence, we created a “context” consisting of the full set of words of the desired part of speech appearing in the document. Weight was assigned to every sense of each word, proportional to the number of occurrences of that word. PPR was run on this context, which we anticipated would concentrate weight in those parts of the tree where multiple weighty synsets reinforced each other, and thus for narrowly focused documents might give a reasonably accurate although approximate estimate of the comparative relevance of different senses of a given word.

For each word in the initial vocabulary, all its WordNet senses were ranked in decreasing order of their PPR-generated weights. The senses whose weights were less than 30% [4] of the highest weight for this word were eliminated; the count of occurrences of this word was then distributed over the remaining senses, proportional to their PPR weights. The Greedy algorithm was then applied to shrink this set.

Weak Disambiguation (PPR-VOC-G) In this algorithm the entire initial vocabulary list was treated as a context for the PPR algorithm. We chose a base set approximately the same size as the starting list of unique words in the document. The PPR weights were multiplicatively scaled to provide the “counts” for this base set. The Greedy algorithm was then applied.

Strong Disambiguation, Top-N (PPR-W2W-G) The strongest context-based disambiguation (PPR-W2W) creates a vocabulary that is then reduced by application of the Greedy algorithm.

[4] Found via parametric testing.

3.2 Evaluation of Reduced Vocabularies

The ideal reduced lexicon would be capable of representing the same concepts as the original vocabulary, but using far fewer words. However, as lexicon size decreases, the semantic precision of expression tends to decrease as well; consequently, measures of the semantic quality of a lexicon are needed to assess our results.

Our primary evaluation of the algorithms is based on a comparison of lexicons from human-simplified text with those we generate algorithmically from non-simplified text. Specifically, articles and simplified versions of articles on the same topics were obtained from the English Wikipedia (EW) and Simple English Wikipedia (SEW) (Wikipedia, 2010). English Wikipedia articles contain a reasonably large sample of standard English written by diverse authors covering a range of topics. Simplified articles are not usually direct simplifications of original articles from the EW: a simplified article generally covers only a subset of the topics covered by the original article using a simpler grammar and lexicon. Consequently, reducing the lexicon of EW articles to that of SEW articles represents a difficult task for any lexical simplification algorithm.

Text and Simplified Text Corpora

All six algorithms were applied to a set of ten articles found in the SEW. We selected articles from the SEW that were listed as “good” simplified articles by managers of Wikipedia, that had at least three paragraphs of prose, and that had a majority of sections that appeared to correspond to sections in articles in the EW. All articles were manually edited to remove images, tables, and references, and to retain only those topics, usually indicated by sections, found in both the original and simplified article. We applied the Stan-

	Chop	City	Evol	Goth	Hum	Mon	Okla	RRH	Sat	Snake
Noun, EW	326	341	817	634	1141	587	817	399	440	225
Noun, SEW	208	136	511	434	390	338	285	228	224	68
Noun, common	145	52	265	295	266	255	232	119	161	34
Verb, EW	161	99	315	221	311	177	207	164	149	127
Verb, SEW	90	43	172	128	119	102	80	85	87	47
Verb, common	61	19	103	73	77	63	43	52	59	23

Table 1: Unique word counts for EW and SEW articles, and the number of words EW and SEW have in common

ford Part of Speech Tagger (Toutanova and Manning, 2000) to isolate nouns and verbs from all documents; resulting noun lexicons from SEW are 30–68% of the size of the EW's vocabulary, while for verb lexicons the corresponding figures are 37–58% (see Table 1). The number of unique nouns in the EW articles varied from 225–1141, with the number of unique nouns in SEW articles ranging from 68–511. Similarly, there were 99-315 unique verbs in the EW articles, and 43-172 unique verbs in the SEW articles.

Comparing the noun and verb counts, we note that the number of unique verbs is about half the number of unique nouns. The verb count varies less across the ten articles, suggesting that, as articles grow longer, they incoporate more additional nouns than additional verbs. Table 1 also indicates the degree of overlap among the noun and verb lexicons for each pair of articles.

Evaluation Measures

Let us call the vocabulary (restricted to a single part of speech) of an EW article L, and the vocabulary of the SEW article on the same topic S. Then we apply one of our algorithms, described above, to L in order to produce a reduced lexicon (or hypernym set) H. We use two measures to assess the quality of the hypernym set. Our first measure, affinity, is used to measure the semantic distance between L and the H generated from it, in order to assess whether expressivity has been adequately retained. The second, Symmetric Vocabulary Distance, is used to measure the semantic distance between the automatically-generated H and S, under the assumption that the vocabulary of the human-simplified text (S) is a reasonable proxy for a human-approved reduction of L. We set the desired size of H to be the size of S, so that we can fairly compare the semantic space covered by H to that covered by S.

Affinity Between Starting Vocabulary and Hypernym Set Intuitively, we aim to generate a precise lexicon, i.e. one in which the semantic distance between the starting vocabulary and the reduced lexicon is minimized; thus the most precise lexicon is the original vocabulary. To operationalize this intuition, we experimented with several distance measures based on path distance in the WordNet tree (Budanitsky and Hirst, 2006) and ultimately adapted a scoring measure proposed in (Widdows, 2003), which finds the distance in the WordNet subtree between a vocabulary word's sense and its nearest ancestor (hypernym) in the lexicon. Widdows calls the inverse square of this distance the *affinity* score for that word. Suppose there are N vocabulary word senses in the base set L, and dist(x) is the distance (number of edges plus one) in the tree from a word sense x to its hypernym in the reduced lexicon H, or ∞ if there is no such hypernym. If c is defined to be the weight (number of occurrences in the document) of the current sense, and C is the summed weight of the entire vocabulary, then our distance measure is defined as:

$$\frac{1}{C} * \sum_{i=1}^{N} \begin{cases} \frac{c}{\text{dist}(i)^2} & \text{if dist}(i) \neq \infty \\ \frac{-c}{4} & \text{if dist}(i) = \infty \end{cases} \qquad (1)$$

Affinity scores increase as distance between synsets decreases.

Symmetric Vocabulary Distance When comparing two vocabularies, an intuitive measure of difference is the semantic distance between a word in the first vocabulary and the word in the second that is semantically nearest to it. This intuition leads to the following definition of vocabulary distance. Let $d(a, b)$ denote a measure of semantic distance between words a and b, here measured by the count of WordNet edges in the shortest path from a to b. Suppose the vocabulary and reduced lexicon are represented as sets S and H, respectively. We define the distance between a word, $s \in S$ and (the entire) lexicon H to be $d(s, H) = min_{h \in H} d(s, h)$. Summing over all the elements of S gives a distance measure that is asymmetric, and we therefore define the symmetric distance between S and H by

$$d(H, S) = \frac{\sum_{s \in S} d(s, H) + \sum_{h \in H} d(h, S)}{2}.$$

4 Evaluation Results

Average affinity scores between vocabulary words in L and their nearest (covering) hypernyms in H are shown in Figure 2. Intervals were generated using bootstrap resampling with 95% confidence. Looking at the results for nouns first, confidence intervals indicate a significant difference between the top two algorithms, Greedy and PPR-W2W-G. For purposes of comparison, we note that these "best" scores are lower than those reported by Widdows (2003) for class labels that correctly classify nouns. In that study Widdows found that affinity scores in the range (0.67, 0.91) were indicative of correct class labels for common nouns, but that affinity scores of about 0.57 indicated incorrect labels. Turning to verbs, again the best results are produced by the PPR-W2W-G algorithm, and again this algorithm appears significantly better than the rest of the algorithms, none of which appear significantly better than the others. In all cases, the results for verbs appear better than the results for nouns; this is at least partly explained by the shallowness of the WordNet verb hierarchy, as compared to the noun hierarchy.

The symmetric distances in Table 3 show the distances between automatically reduced lexicons (H) as compared to the vocabulary from human-simplified text (S). Our algorithms construct lexicons that are on average 1-2 edges away from the manually simplified lexicon; these words are much nearer than those in randomly selected lexicons, which are experimentally measured in lexicons of size 100 to 1000 as about 7 to 4 edges distant for nouns and 5 to 3 edges distant for verbs.

Based on the average results, for both nouns and verbs the best performing algorithm is again PPR-W2W-G. The other algorithms have distance scores similar to those of the frequency-based baseline. Word sense disambiguation appears to improve the precision of the reduced lexicon, though the differences are only sometimes clearly significant. As is the case for the affinity measure, the symmetric vocabulary distances for verbs appear better than for nouns, for every algorithm, in some cases significantly so. Here the difference in the percentage of shared unique verbs (60% on average) vs. shared unique nouns (51% on average) between the EW and SEW articles (L and S) may explain some of this apparent superiority of verb results.

Another way to consider these vocabulary distance results is to compare them to the vocabulary distance of the EW and SEW versions of each article. If the algorithms are successful, we expect the average distance between the reduced lexicons of the full article and the original vocabulary of the human-simplified articles (H and S) to be less than the distance between the original and human-simplified (L and S) lexicons themselves. The average distance between the original full and simplified lexicons (H and S) is 1.97 for nouns and 1.45 for verbs, compared with 1.50 and 1.16 respectively between L and S for the reduced lexicons produced by the best performing algorithm (PPR-W2W-G), a 24% and 12% improvement, respectively.

Examples of Vocabulary Substitution

Two examples of vocabulary replacement for verbs are shown below. We use the evolution article from the EW, which has 458 unique verbs (the largest number among our ten articles). The SEW article on the same topic has 272 unique verbs, and so we produce a reduced lexicon of the same size, a 41% reduction, using the best-performing algorithm, PPR-W2W-G.

In the EW article on evolution, we find the sentence

"IF ONE SPECIES CAN OUT-COMPETE ANOTHER, THIS COULD PRODUCE SPECIES SELECTION, WITH THE FITTER SPECIES SURVIVING AND THE OTHER SPECIES BEING DRIVEN TO EXTINCTION."

Four verbs are marked as such by the W2W word sense disambiguation algorithm; in their

Algorithm	Nouns		Verbs	
	Avg. Distance	95% C.I.	Avg. Distance	95% C.I.
Baseline	0.20	(0.15, 0.26)	0.19	(0.13, 0.26)
Greedy	0.32	(0.23, 0.40)	0.49	(0.46, 0.53)
PPR-N	0.01	(-0.03, 0.06)	0.48	(0.42, 0.54)
PPR-V-G	0.14	(0.11,0.17)	0.51	(0.46, 0.55)
PPR-WSD-G	0.15	(0.13, 0.17)	0.45	(0.39, 0.52)
PPR-W2W-G	0.65	(0.56, 0.72)	0.76	(0.72, 0.80)

Table 2: Average affinities between vocabulary words (from the EW articles) and their nearest hypernyms; higher is better. The averages and 95% confidence intervals are shown for each algorithm.

Algorithm	Nouns		Verbs	
	Avg. Distance	95% C.I.	Avg. Distance	95% C.I.
Baseline	1.93	(1.68, 2.25)	1.35	(1.20,1.52)
Greedy	2.04	(1.77, 2.37)	1.31	(1.22,1.41)
PPR-N	1.85	(1.66, 2.09)	1.58	(1.43,1.74)
PPR-V-G	1.87	(1.63, 2.12)	1.29	(1.20,1.40)
PPR-WSD-G	1.90	(1.63, 2.20)	1.65	(1.47,1.81)
PPR-W2W-G	1.50	(1.33, 1.73)	1.16	(1.04,1.29)

Table 3: Average distances between reduced size noun and verb lexicons, automatically generated from EW articles, and vocabularies extracted from the SEW human-simplified articles on the same topics; lower is better.

root forms they are *produce, survive, be, drive*. The hypernyms selected for each verb by the PPR-W2W-G algorithm are inserted into the sentence below[5]; we manually modified each verb tense to match:

"If one species can out-compete another, this could (produce) species selection, with the fitter species (living) and the other species (being) (moved) to extinction."

An example that did not give very satisfying results is:

"For example, more than a million copies of the Alu sequence are present in the human genome, and these sequences have now been recruited to perform functions such as regulating gene expression."

Here, the verbs that are marked as such are *be, recruit, perform, regulate*. The closest hypernym for *recruit*, *perform*, and *regulate* is the root node we added to the verb graph to make it a tree. The words *recruit* and *regulate* each appears only once in the entire article, and *perform* only twice (both times with the same sense). Words that appear rarely may not be worth inclusion in the reduced lexicon, especially given a very limited lexicon size. One could attempt to automatically detect such cases, considering infrequency of word use and distance from a word to its nearest hypernym.

[5]The hypernyms are synsets rather than words; we simply took the first word from each synset.

Vocabulary Reduction as Simplification

We estimated the difficulty of the various vocabularies derived by the best-performing PPR-W2W-G algorithm by calculating the average Kucera-Francis word frequency over all words in the vocabulary. Multi-word lexical items (e.g., 'get the better of') which are not found in the frequency data were excluded. In addition, words with more than one sense are conflated in the frequency counts and are therefore all senses are treated as a single entry for purposes of measuring frequency. In the case of noun vocabularies, the reduced lexicon found by the PPR-W2W-G algorithm had an average frequency of 123, lying nearer the average frequency of nouns in the EW article (109) than that of the simplified SEW article (153). The average verb frequency score of the reduced lexicon (243) was nearer the simplified score (290) than the wiki score (163). Thus, in both instances the reduced vocabulary consists of more frequently occurring, and thus arguably simpler, words than the original full vocabulary.

5 Discussion

Our results show that the greedy maximization of information can be combined with word sense disambiguation to yield an algorithm for the automated generation of a reduced lexicon for text documents. Among the six algorithms, we found significant differences between an approach that ignores word sense ambiguity and those that address this ambiguity explicitly. The introduction of multiple senses for every word, most of which are unintended in the original document, introduces sense ambiguity that is not overcome by document word counts, even if those counts are readjusted to weight likely senses. Early word sense disambiguation takes advantage of phrase and sentence context (unavailable at later stages of processing) and results in a smaller tree to be searched.

Comparison of the best algorithm (PPR-W2W-G) with the word-frequency Baseline suggests to the relevance of the semantic hierarchy in addition to word-sense disambiguation. The Baseline algorithm uses full word-sense disambiguation, but not the semantic hierarchy of Wordnet. PPR-W2W-G uses both types of semantic information. The superiority of PPR-W2W-G may also be a by-product of evaluation measures that are based on a Wordnet-based definition of semantic distance. In the future, a better test of the relative importance of the two aspects of semantics may be obtained from other judgements of semantic similarity.

One would intuitively expect that lexical simplification is optimally implemented at the level of the document, not the sentence. Although our algorithm does not explicitly select "simple" words for the lexicon, the combined algorithm yields a lexicon in which most words are only 1-2 edges away from human-simplified counterparts. Since our greedy search of WordNet is top-down, our hypernyms lie above the words drawn from the original document; our intuition that these more general words tend to be simpler is supported by the higher average frequency of the resulting lexicons. These results do not offer a complete approach to lexical simplification, but suggest a role for contextualized semantics and appropriate knowledge-based techniques.

6 Acknowledgements

We thank Dr. Barry Wagner for bringing the AAC interface problem to our attention and the Bard Summer Research Institute for student support. Students Camden Segal, Yu Wu, and William Wissemann participated in software development and data collection. We also thank the anonymous reviewers for their suggestions.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*, pages 33–41, Morristown, NJ, USA. Association for Computational Linguistics.

Sven Anderson, S. Rebecca Thomas, Camden Segal, and Yu Wu. 2011. Automatic reduction of a document-derived noun vocabulary. In *Twenty-Fourth International FLAIRS Conference*, pages 216–221.

Bruce Baker. 1982. Minspeak: A semantic compaction system that makes self-expression easier for communicatively disabled individuals. *Byte*, 7:186–202.

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram version 1. Linguistic Data Consortium, Philadelphia.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, March.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.

William Coster and David Kauchak. 2011. Simple english wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 665–669.

Jan De Belder, Koen Deschacht, and Marie-Francine Moens. 2010. Lexical simplification. In *Proceedings of the First International Conference on*

Interdisciplinary Research on Technology, Education and Communication (ITEC 2010), pages 43–63. Cambridge Univ Press.

Henry Kucera and W.Nelson Francis. 1967. *Computational analysis of present-day American English*. Brown University Press, Providence, Rhode Island.

Partha Lal and Stefan Rüger. 2002. Extract-based summarization with simplification. In *Proceedings of the Workshop on Text Summarization at the Document Understanding Conference*.

Tom Richens. 2008. Anomalies in the WordNet verb hierarchy. *Proceedings of the 22nd International Conference on Computational Linguistics - (COLING '08)*, (August):729–736.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 Task 1: English lexical simplification. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 347–355.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70.

Keith Trnka, John Mccaw, Debra Yarrington, Kathleen F McCoy, and Christopher Pennington. 2009. User Interaction with word prediction: The effects of prediction quality. *Association for Computing Machinery Transactions on Accessible Computing*, 1(3):1–34.

Tonio Wandmacher, Jean-Yves Antoine, Franck Poirier, and Jean-Paul Dèparte. 2008. SIBYLLE , an assistive communication system adapting to the context and its user. *Association for Computing Machinery Transactions on Accessible Computing*, 1(1):1–30.

Dominic Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, number June, pages 197–204. Association for Computational Linguistics.

Wikipedia. 2010. Wikipedia, the free encyclopedia. en.wikipedia.org/wiki.

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 365–368.