Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference

Automatic Reduction of a Document-Derived Noun Vocabulary

Sven Anderson and **S. Rebecca Thomas** and **Camden Segal**
Bard College, Annandale-on-Hudson, NY 12504
{sanderso, thomas, cs471}@bard.edu

Yu Wu
Stanford University, Stanford, CA 94305
ywu2@stanford.edu

Abstract

We propose and evaluate five related algorithms that automatically derive limited-size noun vocabularies from text documents of 2,000-30,000 words. The proposed algorithms combine Personalized Page Rank and principles of information maximization, and are applied to the WordNet graph for nouns. For the best-performing algorithm the difference between automatically generated reduced noun lexicons and those created by human writers is approximately 1-2 WordNet edges per lexical item. Our results also indicate the importance of performing word-sense disambiguation with sentence-level context information at the earliest stage of analysis.

Introduction

This paper 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. One motivation for this work stems from the fact that much of the information available on the Web and in other electronic formats is at best weakly structured via its encoding in natural language. Access to this information can be facilitated by artificial intelligence techniques that exploit the semantic relations that bind concepts; however, resources that provide rich representations of semantic relations, such as WordNet (Miller 1995), also have lexicons that are large and have word meanings that are difficult to discriminate. This is especially problematic when large, complex lexicons underlie applications intended for novice users of a language or computer interface.

A variety of techniques estimate adult mental lexicons to contain between 14,000 and 238,000 unique words (Amano and Kondo 1998). Hirsh and Nation have estimated that a vocabulary of approximately 5,000 words suffices to read unsimplified novels intended for teenage native speakers of English (Hirsh and Nation 1992). Tasks for which a smaller, simpler vocabulary is desirable include text summarization, text paraphrase, and interaction with mobile devices or augmentative communication devices.

To expand on the latter task, one type of Augmented and Assistive Communication (AAC) system includes a touchscreen, from which users can select icons that may, alone or in combination, represent specific words or phrases (Baker 1982).[1] Note that 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 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 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.

In the sections that follow, we describe five methods for generating a reduced lexicon from a document-derived vocabulary, describe measures to evaluate the quality of the resulting lexicons, and apply them to the five algorithms. In the final section we interpret our results with respect to other approaches that have similar aims.

Methods

In this initial work, we limit our focus to nouns. Given a starting document in standard English, we extract all and only its nouns using the Stanford Part of Speech Tagger (Toutanova and Manning 2000), and reduce each to its base uninflected form using WordNet's morphstr method. The list of nouns, together with a count of the occurrences of each, is called the *starting vocabulary*. From this, we automatically generate a much smaller *reduced lexicon* and, when materials exist, seek to compare this lexicon to the set of nouns used by a human author who wrote a simplified article on the same topic. We developed and tested five approaches for generating the reduced lexicon.

Personalized PageRank, Top-N (PPR-N)

The most direct algorithm generates a reduced lexicon directly from the results of the Personalized PageRank algorithm (PPR). PPR was originally developed to perform unsupervised word sense disambiguation (Agirre and Soroa

[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).

2009). In PPR-N, for each noun appearing in the starting document, we find the corresponding synset(s) in WordNet and assign them weights proportional to the count of occurrences of the noun. All other synsets are initialized with weight zero. 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.

We ran the implementation of PPR distributed by Agirre and Soroa[2] on this starting configuration, and the N highest-weight synsets in the result comprise the reduced lexicon H. Given any noun in the original vocabulary, we can traverse up the tree from it until we reach a node that is a member of the lexicon; we say this is the lexicon element that *covers* the noun. Note that a lexicon H generated by PPR-N will not necessarily cover every noun.

Greedy Algorithm to Generate a Reduced Lexicon

For our other four approaches, we begin with the starting vocabulary and use it to generate a *base set* of WordNet synsets representing these nouns. The four approaches create a reduced lexicon by applying the same greedy algorithm to a base set; they differ in how the base set is selected. We first describe the greedy algorithm, and then outline the four approaches to base set construction.

We construct a subtree of WordNet containing all the elements of our base set and all of their WordNet ancestors (hypernyms). Note that the WordNet graph for nouns is a tree with the root node containing the word *entity*. We add to each node a count: for leaves, this is the number of occurrences of that word sense in the original document, while for internal nodes, this is the sum of the counts of the node's children plus its own number of occurrences in the document. A hypernym set H is a set of nodes; as for PPR-N, a noun in the base set is said to be *covered* by its nearest ancestor that is in H.

The greedy algorithm simply works its way down the WordNet subtree, at each step greedily adding the "best" child to the growing lexicon; see Figure 1. Note that the sequence of choices of locally "best" children may not lead to a globally optimal solution.

The information gained when a child is added to the tree is computed via the equation

$$-p * \left(\frac{c}{p} \log\left(\frac{c}{p}\right) + \left(\frac{p-c}{p}\right) \log\left(\frac{p-c}{p}\right) \right), \qquad (1)$$

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, the above equation 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 then it is removed from H. Iterations continue until H reaches its target size.

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

Note that if all word occurrences are initially covered by the root node then since nodes are only removed from the H set if they no longer cover any occurrence, it follows that all words in the initial vocabulary are still covered by the resulting H set. In future work, we may wish to relax this requirement, leaving a small number of relatively uncommon words uncovered in order to provide more specific hypernyms for more commonly used words.

Base Set Selection

In generating the base set (of synsets) from the starting vocabulary (of words), the main issue is that many nouns have multiple senses; thus we must have some form of word sense disambiguation (Navigli 2009), and of apportioning the occurrences of a vocabulary word among its senses. The four ways we generate base sets are described next.

No Disambiguation (Greedy alone) Each word's number of occurrences is credited to every one of its senses, so the base set consists of all senses of every noun in the document.

Full PPR-based Word Sense Disambiguation (PPR-W2W-G) PPR was originally intended as a technique for word-sense disambiguation within sentence-like contexts. The version of the PPR algorithm we use here disambiguates one word at a time by concentrating weight on the senses of words occurring near it in the original text. Agirre and Soroa found best disambiguation performance with this method, which they call *w2w* (Agirre and Soroa 2009). In PPR-W2W-G the w2w variant of PPR is first applied to extract a single sense (synset) for each noun occurrence in the document. The base set consists of these synsets.

Estimated Proportions of Word Sense Occurrences (PPR-WSD-G) Full PPR-W2W word sense disambiguation takes from several hours to several days for the documents we studied. In this approach, we avoided doing full disambiguation of each noun occurrence. Instead we created a "context" consisting of the full set of nouns appearing in the document. Weight was assigned to every sense of each noun, proportional to the number of occurrences of that noun. 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.

The base set was then generated as follows: 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% 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. Thus if the word "cat" appeared in the document 100 times, and if PPR assigned weight 10 to the *feline* synset and weight 5 to the *guy* synset (and less than 3 to all other "cat" senses) then the *feline* synset would be assigned a vocabulary word count of 67 and the *guy* synset a count of 33.

Direct Use of PPR Results (PPR-VOC-G) In this approach the entire initial vocabulary list was treated as a context for the PPR algorithm, and the base set simply consisted of the highest-weighted synsets. We chose to let our base set

```
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 by adding c to H
    select the child, x, with the maximal information gain; insert it 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

be approximately the same size as the starting list of unique nouns in the document. The PPR weights were multiplicatively scaled to provide the "counts" for this base set.

Once the base set is generated, by whichever method, the tree is constructed and the greedy algorithm of Figure 1 is used to generate the hypernym set, as described earlier.

Evaluation of Lexicons

The algorithms described generate reduced lexicons that are designed to be much smaller than the lexicon of the original document. Although the algorithms also apply to the verbs, adjectives, and adverbs found in WordNet, those WordNet structures are less studied and in some cases do not have tree structure. We therefore limited our initial study to nouns in order to focus on evaluation within a single well-developed part of the WordNet knowledge base. 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. Articles and simplified versions of articles on the same topics were obtained from the English and Simple English Wikipedia (Wikipedia 2010). Simplified articles are not direct simplifications of original articles from the English Wikipedia, but a simplified article generally covers a subset of those topics covered by the original article.

Text and Simplified Text Corpora

All five algorithms were applied to a set of ten articles (Table 1) found in the Simple English Wikipedia. We selected articles from the Simple English Wikipedia that were listed as "good" simplified articles by managers of Wikipedia and that had a majority of sections that appeared to correspond to sections in articles in the English Wikipedia. In addition, we selected articles that had at least three paragraphs of prose. All articles were edited to remove images, tables, and references. Only those topics, usually indicated by sections, found in both the original and simplified article were retained. We applied the Stanford Part of Speech Tagger (Toutanova and Manning 2000) to isolate nouns from all documents; resulting noun lexicons from simplified text are 30-71% the size of the original document's noun vocabulary, as shown in Table 2.

Identifier	Article
A	Snake
B	City
C	Chopsticks
D	Red Riding Hood
E	Monarch (butterfly)
F	Saturn (planet)
G	Gothic (architecture)
H	Oklahoma (state)
I	Human
J	Evolution

Table 1: Dataset articles from the English Wikipedia and Simple English Wikipedia.

Evaluation Measures

Affinity Between Lexicon Entry and Hypernym In Reduced Lexicon Intuitively, we aim to generate a precise lexicon, i.e. one in which the semantic distance between vocabulary items and the reduced lexicon is minimized. (Thus the most precise lexicon is the original vocabulary.) To operationalize this, we experimented with a number of distance measures based on path distance in the WordNet tree 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, 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, 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 lexicon, then our distance measure is defined as:

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

For those algorithms in which no disambiguation is performed, we calculate the score for each sense of a word and average these scores. When disambiguation is performed, the base set consists of synsets and there is no ambiguity about the relevant sense. If a word or synset has no ancestor in the reduced lexicon, a penalty of -0.25 is added to the score as shown in Equation 2. Affinity scores increase as distance between synsets decreases.

Distance Between Vocabulary and Reduced Lexicon When comparing a lexicon and a vocabulary, an intuitive measure of difference is the semantic distance between a word in the vocabulary and the nearest word in the lexicon. This intuition leads to the following definition of lexi-

	Article ID									
	A	B	C	D	E	F	G	H	I	J
Sentences, wiki	81	86	140	110	177	183	269	224	374	421
Sentences, smpl-wiki	31	61	95	108	186	133	330	139	200	315
Nouns, wiki	448	585	757	770	1084	1152	1716	2109	2861	2983
Nouns, smpl-wiki	103	387	428	371	675	656	1509	621	888	1328
Unique Nouns, wiki	223	312	312	555	538	472	651	814	1290	908
Unique Nouns, smpl-wiki	68	141	212	226	321	249	462	287	427	566
Reduced Lexicon (%)	30	45	68	41	60	53	71	35	33	62

Table 2: Summary statistics for original and human-simplified articles from Wikipedia, listed in order of increasing vocabulary size. Article titles corresponding to IDs are listed in Table 1. All rows are counts except the final row, which indicates ratio of unique nouns in the simplified article to unique nouns in the full article.

con distance. Let $d(a, b)$ denote a measure of semantic distance between words a and b, in this report 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 V and L, respectively. We define the distance between a word, $v \in V$ and (the entire) lexicon L to be $d(v, L) = min_{w \in L} d(v, w)$. This distance measure is asymmetric, and we therefore define the distance between V and L as follows:

$$d(L, V) = \frac{\sum_{v \in V} d(v, L) + \sum_{l \in L} d(l, V)}{2} \quad (3)$$

In a finite graph such as WordNet the distance between sets of selected nodes decreases as the proportion of nodes selected in the tree grows. The lexicon distance measure $d(L, V)$ will therefore decrease as the size of the lexicon grows. This relationship was verified by generating random lexicons from the 82,144 different noun synsets in WordNet 3.0 and then measuring the distance between the lexicons using Equation 3. Figure 2 shows how the distance between lexicons varies inversely with vocabulary size: the measure decreases from about 8 to 3 as lexicon size increases from 20 to 4700 words, the size of noun vocabularies in this report. This range establishes an inter-lexicon distance our automatically generated lexicons should fall below; we certainly want to do better than a randomly generated lexicon!

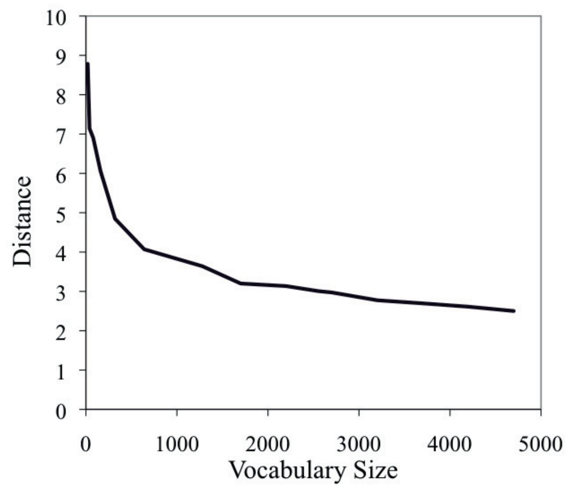

Figure 2: Distance (number of edges) between two random noun lexicons, both of size N, as N is increased from 20 to 4700.

Evaluation Results

Average affinity scores between vocabulary words and their nearest hypernyms are shown in Table 3. Intervals were generated using bootstrap resampling with 95% confidence. 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 (Widdows 2003) for class labels that correctly classify nouns. In that study Widdows found that high 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.

We also explored how the size of the reduced lexicon affects the affinity score; to give richer data, we used three book-length documents dealing with agriculture, dog breeding, and cooking, respectively. The results for the PPR-W2W-G algorithm are shown in Figure 3. As expected, the affinity increases with the lexicon’s size, since the lexicon can encompass an increasing proportion of the initial vocabulary. In order to reach the affinity of 0.67 proposed by Widdows, the reduced lexicon should be at least 40-50% of the size of the original noun set. The affinity score does not reach 1.0 because a few nouns, such as proper nouns, are not found in WordNet.

The lexicon distance measure provides a direct measure of distance in WordNet for two vocabularies. The scores in Table 4 list the distances between automatically reduced lexicons and the vocabulary from human-simplified text. Most 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 as about 7 to 4 edges distant in lexicons of size 100 to 1000, respectively. Based on the average results, the best performing algorithm is PPR-W2W-G,

Algorithm	Article ID										Avg.	95% C.I.
	A	B	C	D	E	F	G	H	I	J		
Greedy	0.05	0.23	0.26	0.28	0.33	0.30	0.51	0.35	0.33	0.52	0.32	(0.23,0.40)
PPR-N	0.00	0.00	-0.07	0.02	-0.04	-0.11	0.00	0.13	0.09	0.11	0.01	(-0.03,0.06)
PPR-V-G	0.07	0.13	0.12	0.12	0.12	0.10	0.14	0.22	0.17	0.20	0.14	(0.11,0.17)
PPR-WSD-G	0.12	0.14	0.12	0.13	0.14	0.11	0.18	0.21	0.19	0.18	0.15	(0.13,0.17)
PPR-W2W-G	0.36	0.57	0.74	0.74	0.72	0.63	0.83	0.56	0.55	0.79	0.65	(0.56,0.72)

Table 3: Average affinities between vocabulary words and their nearest hypernyms; higher is better. These results are for the full, unsimplified Wikipedia articles. The averages and 95% confidence intervals are shown for each algorithm.

Algorithm	Article										Avg.	95% C.I.
	A	B	C	D	E	F	G	H	I	J		
Greedy	2.58	3.08	2.17	2.34	1.75	1.91	1.75	1.85	1.52	1.48	2.04	(1.77, 2.37)
PPR-N	1.94	2.57	2.03	2.09	1.62	1.79	1.51	2.00	1.40	1.55	1.85	(1.66, 2.09)
PPR-V-G	1.72	2.68	2.21	2.19	1.73	1.85	1.59	1.85	1.25	1.60	1.87	(1.63, 2.12)
PPR-WSD-G	2.30	2.92	2.10	2.15	1.82	1.81	1.61	1.54	1.26	1.54	1.90	(1.63, 2.20)
PPR-W2W-G	2.18	1.99	1.48	1.64	1.23	1.26	1.32	1.43	1.33	1.17	1.50	(1.33, 1.73)

Table 4: Distances between reduced size noun lexicons, automatically generated from full Wikipedia articles, and noun vocabularies extracted from the Simple English Wikipedia human-simplified articles on the same topics; lower is better.

which uses sentence contexts to disambiguate words prior to creating the lexicon via the greedy algorithm. Word sense disambiguation appears to improve the precision of the reduced lexicon in all cases, though it is only clearly significant when comparing PPR-W2W-G and the Greedy algorithm.

Histograms of lexicon distance scores show that for most of the algorithms, about 40% of the synsets for human-simplified and automatically derived lexicons are identical. A representative example, the histograms of distance scores for the Evolution article for each of the five algorithms, is shown in Figure 4. Histograms for the other articles are quantitatively similar.

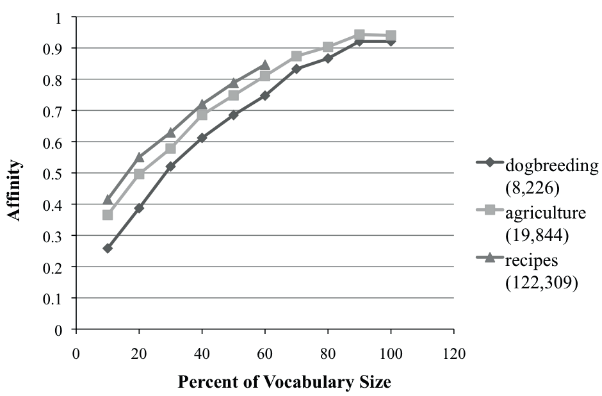

Figure 3: Affinity scores as a function of reduced vocabulary size for three large documents. The total number of noun occurrences in the document is indicated in parentheses.

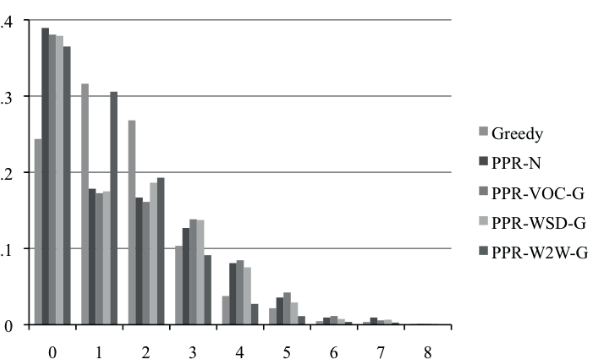

Figure 4: Histogram of distance scores between reduced lexicon and vocabulary of human-simplified texts for the Evolution article.

Conclusion

Among the five algorithms, we found significant differences between an approach that ignores word senses and those that incorporate them explicitly. The introduction of numerous 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 most likely senses. Early word sense disambiguation takes advantage of phrase and sentence context (unavailable at later stages of processing) which also results in a smaller tree to be searched.

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. 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 the WordNet noun taxonomy is top-down, this suggests that many simplified words lie above nouns from the original unsimplified version and are therefore more general than those in the original document.

Like some approaches to labeling, semantic tagging and paraphrase, our algorithm relies on a lexical knowledge base and its interaction with a text corpus. Green and Dorr note that much of the difficulty with the paraphrase problem is due to the fact that the same semantic content can be presented in numerous ways (Green and Dorr 2004). They propose a system, SemFrame, that is similar to this project in its overall goal of producing a reduced-size noun set from an initial set, and in its approach of finding covering nodes in the WordNet noun tree. SemFrame generates names for slots in semantic frames; starting with a collection of (a few tens of) nouns semantically related to a particular verb sense frameset, it reduces this set to a few possible frame slot descriptions. Their work uses a measure of conceptual density to select WordNet nodes to retain (Agirre and Rigau 1996); a similar role is played by Eq. 1 in our approach. SemFrame's frame names were evaluated by human judges; the results are not directly comparable to those we report.

A second task closely related to lexical reduction is the semantic tagging of text with a relatively small set of class labels. The objective of this work is the creation of a set of class labels (i.e., a reduced lexicon) that covers a vocabulary while minimizing ambiguity (Cucchiarelli and Velardi 1997). Rather than pursuing a linguistically motivated definition of lexical ambiguity, our approach, influenced by the distance measures examined by Budanitsky and Hirst (Budanitsky and Hirst 2006), seeks to maximize information, leaving linguistic concerns to a later stage. We anticipate that the inclusion of linguistic information (e.g., measures of word difficulty) will guide our future work with text simplification.

The algorithm presented here is not directly applicable to verbs as they are represented in WordNet. The most significant impediment is that the graphs for verbs do not have a common unique ancestor node, but have at least 15 different head nodes and hundreds of verbs with no unique ancestor (Richens 2008). We are currently exploring the application of our approach to this more problematic word category.

A goal of our larger research project is to use a reduced lexicon to select icons for an AAC interface. The mapping from a reduced lexicon to an icon set may be complex, but we initially plan to test a mapping that generates each icon from a partition of the reduced vocabulary. Current AAC systems use expert-selected icon sets that could be dynamically reconfigured to select a small set of the most useful icons if a means for automatically generating icon sets were found.

Acknowledgements

We thank Dr. Barry Wagner for bringing the AAC interface problem to our attention and the Bard Summer Research Institute for student support.

References

Agirre, E., and Rigau, G. 1996. Word sense disambiguation using conceptual density. In *Proc. 16th Conf. on Comp. Ling.*, 16–22. Morristown, NJ, USA: ACL.

Agirre, E., and Soroa, A. 2009. Personalizing PageRank for word sense disambiguation. In *Proc. 12th European Chapter of ACL.*

Amano, S., and Kondo, T. 1998. Estimation of mental lexicon size with word familiarity database. In *Fifth International Conference on Spoken Language Processing*.

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

Budanitsky, A., and Hirst, G. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.* 32(1):13–47.

Cucchiarelli, A., and Velardi, P. 1997. Automatic selection of class labels from a thesaurus for an effective semantic tagging of corpora. In *Proc. of the Fifth Conference on App. NLP*, 380–387. Morristown, NJ, USA: ACL.

Green, R., and Dorr, B. 2004. Inducing a semantic frame lexicon from WordNet data. In *TextMean '04*, 65–72. Morristown, NJ, USA: ACL.

Hirsh, D., and Nation, P. 1992. What vocabulary size is needed to read unsimplified texts for pleasure? *Reading in a foreign language* 8:689–689.

Miller, G. A. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38(11):39–41.

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.* 41(2):1–69.

Richens, T. 2008. Anomalies in the WordNet verb hierarchy. In *Proc. 22nd Int. Conf. on Comp. Ling.*, 729–736. ACM.

Toutanova, K., and Manning, C. D. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *EMNLP/VLC-2000*, 63–70.

Trnka, K.; McCaw, J.; Yarrington, D.; McCoy, K. F.; and Pennington, C. 2009. User interaction with word prediction: The effects of prediction quality. *ACM Trans. Access. Comput.* 1(3):1–34.

Wandmacher, T.; Antoine, J.-Y.; Poirier, F.; and Départe, J.-P. 2008. Sibylle, an assistive communication system adapting to the context and its user. *ACM Trans. Access. Comput.* 1(1):1–30.

Widdows, D. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *NAAC-HLT*, 197–204. Morristown, NJ, USA: Assoc. for Comp. Ling.

Wikipedia. 2010. Wikipedia, the free encyclopedia. [Online; accessed 16-September-2010].