Non-Syntactic Word Prediction for AAC

Karl Wiegand
Northeastern University
360 Huntington Ave
Boston, MA 02115, USA
wiegand@ccs.neu.edu

Rupal Patel, Ph.D.
Northeastern University
360 Huntington Ave
Boston, MA 02115, USA
r.patel@neu.edu

Abstract

Most icon-based augmentative and alternative communication (AAC) devices require users to formulate messages in syntactic order in order to produce syntactic utterances. Reliance on syntactic ordering, however, may not be appropriate for individuals with limited or emerging literacy skills. Some of these users may benefit from unordered message formulation accompanied by automatic message expansion to generate syntactically correct messages. Facilitating communication via unordered message formulation, however, requires new methods of prediction. This paper describes a novel approach to word prediction using semantic grams, or "sem-grams," which provide relational information about message components regardless of word order. Performance of four word-level prediction algorithms, two based on sem-grams and two based on n-grams, were compared on a corpus of informal blogs. Results showed that sem-grams yield accurate word prediction, but lack prediction coverage. Hybrid methods that combine n-gram and sem-gram approaches may be viable for unordered prediction in AAC.

1 Introduction

Many individuals with severe speech impairments rely on augmentative and alternative communication (AAC) devices to convey their thoughts and desires. Those with limited or emerging literacy skills may use icon-based systems, which often require that vocabulary items be selected in syntactic order to generate syntactically well-formed messages; however, selecting vocabulary items serially and in syntactic order can be physically and cognitively arduous depending on the icon organization scheme (Udwin and Yule, 1990). Moreover, AAC productions are often syntactically incomplete or incorrect (Van Balkom and Welle Donker-Gimbrere, 1996), perhaps for efficiency or due to limited linguistic abilities. For many users, unordered vocabulary selection may alleviate the physical and cognitive demands of message formulation and shift the onus of generating syntactically complete and accurate messages onto the AAC device. Although unordered message formulation schemes have been proposed (Karberis and Kouroupetroglou, 2002; Patel et al., 2004) and techniques have been developed for expanding incomplete input (McCoy et al., 1998), prediction has not been incorporated. This paper presents an initial step toward text prediction from a set of unordered vocabulary selections.

Rate enhancement is a commonly cited issue in AAC because aided message formulation rates are an order of magnitude slower than spoken interaction (Beukelman and Mirenda, 1998). Prediction is a common rate enhancement technique. Text prediction for AAC has primarily focused on well-ordered, syntactic input and has leveraged both semantic characteristics (Demasco and McCoy, 1992; Li and Hirst, 2005; Nikolova et al., 2010) and variations of n-grams (Lesher et al., 1998; Trnka et al., 2006). For example, semantic networks and linguistic rules have been used to predict missing function words and to apply affixes to content words (McCoy et al., 1998). The use of n-grams to predict text entry has been extensively studied at both the level of

NAACL-HLT 2012 Workshop on Speech and Language Processing for Assistive Technologies (SLPAT), pages 28–36,
Montréal, Canada, June 7–8, 2012. ©2012 Association for Computational Linguistics

letters (Broerse and Zwaan, 1966; Suen, 1979; How and Kan, 2005) and words (Bickel et al., 2005). For example, memory based language models have been used to predict missing content words using trigrams (Van Den Bosch, 2006). Although some recent work has attempted to loosen syntactic requirements by including either left or right context, some directional context has historically been required (Van Den Bosch and Berck, 2009). Furthermore, word prediction approaches in AAC have typically been implemented for letter-by-letter message formulation (Koester and Levine, 1996; Koester and Levine, 1997; Lesher and Rinkus, 2002; Higginbotham et al., 2009). The current work is fundamentally novel in that: (1) no syntactic order is implied or required during either training or testing; and (2) the prediction is implemented at word level to accommodate icon-based interaction.

Previous work in information retrieval has explored relationships between words with regard to distance (Lin and Hovy, 2003; Lv and Zhai, 2009), grammatical purpose (Tzoukermann et al., 1997; Allan and Raghavan, 2002), and semantic characteristics (Westerman and Cribbin, 2000; Fang and Zhai, 2006; Hemayati et al., 2007), particularly for retrieving highly relevant documents or passages. One study in this area resulted in an approach called s-grams, a generalization of n-grams, in which the distance between words directly affects the strength of their semantic relationship (Järvelin et al., 2007). Another approach to predicting semantically related words is to use collocation to indicate topic changes within a moving window of fixed length (Matiasek and Baroni, 2003). Rather than relying on distance to indicate relationship strength, the current work combines frequency analysis with syntactic indications of semantic coherence.

1.1 Semantic Grams

Semantic grams, or "sem-grams," provide an alternative approach to quantifying the relationship between co-occurring words. A sem-gram is defined as a multiset of words that can appear together in a sentence (Table 1). In English, a sentence is one of the smallest units of language that is typically both coherent, in terms of semantic content, and cohesive, in that the semantic content is inter-related. Additionally, because sentences are demarcated with syntactic cues such as punctuation, semantically related items can be efficiently identified using sentence boundary detection (Kiss and Strunk, 2006). Thus, sem-grams leverage sentence-level co-occurrence to extract semantic content at different levels of granularity, depending on the allowable lengths of multisets. Sem-grams can be viewed as non-directional s-grams with a uniform weight applied to all relationships between any words in a given sentence.

Table 1: Example of Sem-Grams of Length 2

Sentence: "I like to play chess with my brother." **Filtered Words:** i, like, play, chess, brother	
Sem-grams and Counts:	
brother, chess (1)	brother, i (1)
brother, like (1)	brother, play (1)
chess, i (1)	chess, like (1)
chess, play (1)	i, like (1)
i, play (1)	like, play (1)

In a sentence of length L (in words), the number of n-grams of length n (in words), where $L \geq n$, is given by the expression $L - n + 3$, which includes the beginning and ending n-grams that contain null elements. By contrast, the number of sem-grams of length n is given by the expression $\binom{L}{n}$. Thus, there will typically be many more sem-grams of length n in a single sentence than n-grams of the same length. Unlike n-grams, it is not necessary for sem-grams to contain null elements because a sem-gram of length S with a null element is equivalent to a sem-gram of length $S - 1$ without null elements. Sem-grams of length one, containing a single word, are equivalent to the prior probability of that word.

1.2 Prediction Algorithms

Unordered word prediction poses the following problem: given a multiset of existing words E that have already been selected by a user and a set of candidate words C that the user may select from, which candidate word $c \in C$ is the user most likely to select in order to complete the message? As an initial step toward addressing this problem, the following four algorithms, two based on sem-grams and two based on n-grams, were compared:

S1: Naive Bayesian Sem-grams Given existing words E, rank all candidate words $c \in C$ in descending order of probability according to:

$$P(c|E) = P(c) \prod_{w \in E} P(w|c)$$

S1 is a modification of the Bayesian ranking of sem-grams in that it assumes independence of existing words to each other, conditional on the given candidate word. Using true Bayesian probabilities for sem-grams, the probability of a candidate word could be represented as the following for each $P(c|E)$, given $w \in E$ and $|E| = 3$:

$$\frac{P(c)P(w_1|c, w_2, w_3)P(w_2|c, w_3)P(w_3|c)}{P(w_1, w_2, w_3)}$$

The exact form of this equation depends on the ordering branch chosen, but it also requires joint probabilities for sem-grams of different lengths. Assuming conditional independence of the existing words to each other, S1 only requires sem-grams of length two.

S2: Independent Sem-grams Given existing words E, rank all candidate words $c \in C$ in descending order of probability according to:

$$P(c|E) = \prod_{w \in E} P(w, c)$$

The approach of S2 is a "hand of cards" approach that treats the message formulation task as a random drawing of sem-grams from a pool. While the formula above is specified for sem-grams of length 2, it can be extended to support sem-grams of any length.

N1: Naive Bayesian N-grams Given existing words E, rank all candidate words $c \in C$ in descending order of probability according to:

$$P(c|E) = P(c) \prod_{w \in E} P(w|c)$$

N1 is a copy of S1, except that the definition of the joint probability $P(w, c)$ includes the counts for n-grams that contain both w and c, regardless of order. This algorithm was designed to compare whether the information provided by n-grams can be used to approximate the information provided by sem-grams. N1 assigns high ranks to candidate words that are likely to appear adjacent to all other words in the sentence.

N2: Applied N-grams Given existing words E, rank all candidate words $c \in C$ in descending order of probability according to:

$$P(c|E) = \sum_{w \in E} P(w, c)$$

N2 is designed to leverage the strength of n-grams and rank candidate words based on the probability of them appearing adjacent to any of the existing words. N2 uses the same definition of joint probability as N1, where $P(w, c)$ includes the counts for n-grams that contain both w and c, irrespective of order.

2 Method

2.1 Corpus Selection and Preparation

Given the lack of large corpora of AAC message formulations (Lesher and Sanelli, 2000), approximations have often been used (Wandmacher and Antoine, 2006; Trnka and McCoy, 2007). Despite recent efforts to create AAC-like corpora (Vertanen and Kristensson, 2011), statistical prediction is often more effective with larger data sets. The Blog Authorship Corpus (Schler et al., 2006) was selected because it is freely available and tends to be written in an informal style, such as might be seen in diary entries or personal emails. The corpus is both large and diverse, comprising over 140 million words written by 19,320 bloggers in August 2004. The bloggers ranged in age from 13 - 48 and were equally divided between males and females.

To prepare the corpus, all blog posts were extracted as ASCII text. Every blog post was split into sentences using the PunktSentenceTokenizer (Kiss and Strunk, 2006) of the Natural Language Toolkit (NLTK) (Bird et al., 2009) and then split into words using the following regular expression:

```
\w+(\w*([\-\'\.]\w+)*)*
```

English stop words were removed according to a popular list (Ranks, 2012) and remaining words were stemmed using the NLTK's PorterStemmer, which is a modified implementation of the original Porter stemming algorithm (Porter, 1997). Finally, all stemmed words were examined for membership in a stemmed American-English dictionary (Ward,

Table 2: Sample Test Results for N1 and S1

Original Sentence: “but i went to church yesterday with the fam.”
Target Stem: went
Input Stems: yesterday, church
N1 Candidate List: went, morn, today, go, attend, work, afternoon, church, got, day, back, ...
S1 Candidate List: went, go, church, today, got, day, like, time, just, well, one, get, peopl, ...

Original Sentence: “You never see signs like that in cities.”
Target Stem: like
Input Stems: never, see, sign, citi
N1 Candidate List: just, show, sign, realli, say, want, go, seen, thought, hall, citi, live, ...
S1 Candidate List: never, will, like, can, go, love, one, just, know, want, get, live, time, ...

Original Sentence: “This semester Im taking six classes.”
Target Stem: class
Input Stems: take, semest, six
N1 Candidate List: next, month, class, hour, last, second, week, year, first, five, flag, ...
S1 Candidate List: class, month, year, last, time, one, go, day, get, school, will, first, ...

Original Sentence: “Hey, they’re in first, by a game and a half over the Yankees.”
Target Stem: game
Input Stems: yanke, hey, first, half
N1 Candidate List: game, stadium, like, hour, time, year, day, guy, hey, fan, say, one, two, ...
S1 Candidate List: game, got, like, red, time, play, team, sox, hour, go, fan, one, get, day, ...

Note: Uncommon spelling (e.g. semest) is due to stemming.

2002). Any stemmed words not found in the dictionary were removed to further constrain the vocabulary and account for spelling errors and nonsensical text.

The corpus was then randomly split into a training and testing set based on authorship, with 80% of the authors (15,451) being placed in the training set and 20% of the authors (3,871) being placed in the testing set. The training set comprised over 7 million sentences written by 7,682 males and 7,768 females with a combined average age of 22 years. All n-gram and sem-gram statistics, with plus-one smoothing, were gathered using only sentences in the training set and both n-grams and sem-grams were limited to a word length of 2 (bigrams).

2.2 Evaluation

Testing was conducted on 2,000 sentences that were randomly selected from the test corpus. The same processing steps used during training were performed on the test sentences: stop words were removed, the remaining words were stemmed, and all stems not in the dictionary were filtered out. To avoid run-on sentences and sentence boundary detection errors, all test sentences were also truncated to a maximum of 20 words. The words in each test sentence were then shuffled and one word was removed at random and designated as the target word. Each of the four algorithms were provided the shuffled words as input; as output, each algorithm attempted to identify the target word by generating a ranked list of candidates (Table 2).

In addition to the shuffled multiset of input words, each algorithm required a seed list of candidate words. Ideally, all known words in the corpus would be used as candidate words. To constrain the computational requirements, the two algorithms based on n-grams (N1 and N2) were provided with the list of most frequently co-occurring words that appeared as n-grams with any of the multiset of input words, limited to the top 10 n-grams for a given input word. Similarly, each sem-gram algorithm (S1 and S2) received a list of most frequently co-occurring words that appeared as sem-grams with any of the multiset of input words, limited to the top 10 sem-grams for a given input word. With a limit of 19 input words (20 minus the target word), each algorithm received at most 190 unique candidate words to rank.

Two evaluation metrics were used to quantify the performance of each algorithm: (1) a boolean value that was true if the output list contained the target word in any position, indicating that the target word had been successfully predicted; (2) if the algorithm successfully predicted the target word, the algorithm received a positive integer score corresponding to the position of the target word in the output list, with lower scores indicating more accurate prediction. For example, if an algorithm suggested the target word as the first item in its ranked list, it received a score of 1; if it suggested the target word as the second item in its ranked list, it received a score of 2. For computational convenience, the output lists of each algorithm were truncated to the first 100 items; thus, if an algorithm's output list contained the target word in a position after 100, it was marked as failing to predict the target word.

3 Results

The n-gram algorithms successfully predicted 32% of the 2,000 test sentences while the sem-gram algorithms successfully predicted 22% (Table 3). Although both n-gram algorithms performed similarly, N1 consistently predicted the target word more accurately than N2. On average, N1 suggested the target word as the 16th word in its ranked list, where N2 suggested the target word as the 20th word in its list. While the sem-gram algorithms predicted fewer sentences than the n-gram algorithms, they were almost twice as accurate on sentences that they did predict. On average, S1 suggested the target word as the 9th word in its ranked list; for S2, the target word was the 13th item.

To further compare the effectiveness of sem-grams and n-grams, sentences were grouped according to their input length, from 1 to 19 words, and statistics were gathered for each algorithm on each sentence length (Table 4). For test sentences in which the algorithms were only given a single input word, both n-gram algorithms ranked the target word at least one full ranking higher than either sem-gram algorithm, thus giving more accurate predictions. For all other sentence lengths, the sem-gram algorithms were more accurate. Between the n-gram algorithms, N1 consistently predicted the target word more accurately and more often than N2. Similarly, S1 consistently predicted the target word more accurately and more often than S2.

Table 3: Summary of Results

	N1	N2	S1	S2
Sentences	2000	2000	2000	2000
# Predicted	647	649	435	435
% Predicted	32%	32%	22%	22%
Avg Score	16.26	19.70	9.04	12.67

For every input sentence length greater than one, S1 outperformed N1 in all gathered metrics. When comparing the prediction accuracy of N1 and S1, S1's prediction accuracy was also more stable, with N1's prediction accuracy continuing to degrade as the length of the input sentence increased (Figure 1).

4 Discussion

Message formulation using AAC devices has historically relied on serial selection of letters or words (icons). To produce syntactically correct messages for icon-based AAC, selection is often required to proceed in syntactic order. The current work aimed to facilitate unordered vocabulary selection through the use of text prediction. Results indicate that word prediction for unordered message formulation is viable using statistical approaches. Although the n-gram algorithms predicted a larger number of test sentences than the sem-gram algorithms, evaluation of the ranked output indicated that the sem-gram approaches were more accurate. Because n-grams assume that adjacent words are strongly related, it was expected that n-grams would provide more accurate prediction for shorter sentences; however, this advantage was not maintained as sentence length increased beyond two words. Prediction accuracy is likely to be more important in AAC devices because the cognitive demands of choosing from prediction lists can sometimes outweigh rate enhancements (Koester and Levine, 1996; Koester and Levine, 1997).

The use of bigrams may have resulted in poor accuracy of the n-gram algorithms because there were many more sem-grams than n-grams of length 2. Increasing n-gram length, up to a cardinality equal to the number of sem-grams of length 2, could allow n-

Table 4: Prediction Coverage (%) and Average Scores by Sentence Length

# Words	N1 %	N1 Avg	S1 %	S1 Avg	N2 %	N2 Avg	S2 %	S2 Avg
1	20.88%	3.44	12.05%	4.47	20.88%	3.42	12.05%	4.47
2	26.55%	6.07	19.47%	5.89	26.55%	6.32	19.47%	6.23
3	22.22%	7.64	16.89%	6.87	22.22%	9.82	16.89%	9.84
4	32.11%	10.46	22.94%	7.62	32.11%	11.91	22.94%	9.94
5	31.25%	12.13	21.88%	6.14	31.25%	14.02	21.88%	9.14
6	38.18%	15.25	26.67%	8.75	38.18%	17.68	26.67%	12.11
7	42.86%	16.17	29.46%	9.52	42.86%	21.77	29.46%	12.73
8	39.60%	18.08	25.74%	11.15	39.60%	22.00	25.74%	15.73
9	29.11%	19.13	20.25%	11.31	29.11%	23.48	20.25%	17.88
10	44.74%	24.47	35.53%	10.52	44.74%	23.56	35.53%	16.22
11	38.46%	28.55	26.92%	15.21	38.46%	26.80	26.92%	17.93
12	46.00%	23.39	14.00%	13.71	46.00%	41.26	14.00%	9.14
13	38.46%	24.47	25.64%	14.30	38.46%	34.07	25.64%	15.90
14	29.41%	26.30	14.71%	10.80	29.41%	39.10	14.71%	26.20
15	46.67%	32.14	20.00%	16.17	46.67%	36.79	20.00%	15.17
16	47.62%	25.70	28.57%	12.83	47.62%	30.50	28.57%	12.67
17	53.85%	23.14	38.46%	12.20	53.85%	35.14	38.46%	21.40
18	40.95%	38.35	25.71%	13.56	42.86%	43.07	25.71%	25.11
19	38.46%	23.80	38.46%	11.00	38.46%	52.40	38.46%	32.00

gram algorithms to potentially match or surpass the prediction accuracy of sem-grams. For unordered word prediction, this larger set of n-grams would need to be indexed in an order-independent manner, which would further increase computational demands. Such prediction lags, however, are unlikely to be tolerated by users as they engage in interactive tasks (Higginbotham et al., 2009).

Of the two n-gram algorithms, N1 outperformed N2 on both prediction coverage and accuracy. It was hypothesized, however, that N2 would yield more accurate predictions because the target word was defined to be adjacent to at least one of the input words. It was expected that N1 would unfairly reward candidate words that had appeared adjacent to each input word in the training set, while punishing more desirable candidate words that had not appeared adjacent to some of the input words. Perhaps this bias was not evident in the current corpus because plus-one smoothing removed all zero probabilities for adjacency likelihoods. Additionally, N1 may have been more successful because it favored candidates that were related to all input words rather than candidates that were strongly related to just a subset of the input words.

Despite the encouraging prediction coverage of n-grams and the prediction accuracy of sem-grams, approximately two-thirds of the test sentences were not predicted by any of the algorithms. One possible explanation may relate to the decision to seed each algorithm with only the top 10 most frequent words that co-occurred with each input word. Ideally, each algorithm would have considered all words in the vocabulary as candidate words; however, because there were almost 40,000 unique stems in the processed corpus, the computational requirements were prohibitive for this initial implementation. An open empirical question is whether increasing the seed values to include a larger set of co-occurring words would result in greater prediction coverage. It should be noted, however, that while seeding sem-grams with more candidate words may improve prediction coverage, it is unlikely to increase prediction accuracy for the n-gram approaches.

Icon-based AAC devices typically have active vocabularies with much fewer than 40,000 words, which may negate the need for seeding candidate words. For example, two commonly used icon

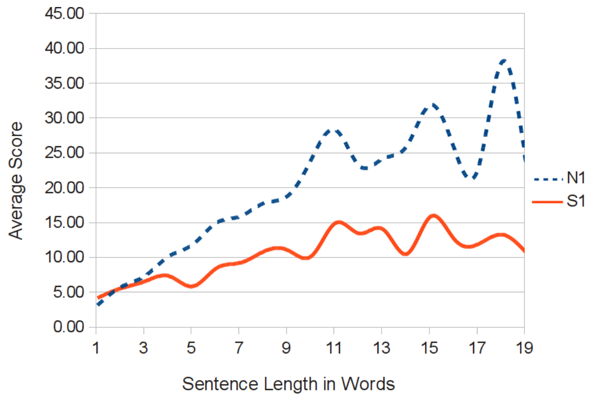

Figure 1: Average score per sentence length for both N1 and S1 (lower scores indicate more accurate prediction).

sets, the Widgit Symbol Set and the Mayer-Johnson Picture Communication Symbol collection, each contain approximately 11,000 icons (Widgit, 2012; Mayer-Johnson, 2012). While a large dictionary was used in this work to provide a conservative estimate of prediction performance, it is possible that using a smaller and more representative AAC vocabulary would improve prediction coverage and accuracy. Additionally, restricting vocabulary size would also reduce computational demands, making it more feasible to use all vocabulary words as candidates.

5 Conclusion and Future Directions

The current work provides a promising approach to word prediction for AAC users who may benefit from unordered message formulation. Sem-grams make use of co-occurrence between words within a sentence to improve prediction accuracy. While n-grams have historically provided a strong foundation for word prediction in letter-by-letter systems, results indicate that they can also be used for unordered word prediction, although they are not as accurate as sem-grams. A hybrid approach that seeds both types of algorithms with a superset of candidate words and merges the prediction lists may simultaneously exhibit the wide prediction coverage of n-grams and the high prediction accuracy of sem-grams. Such a hybrid approach could enhance the speed of unordered message formulation and increase social engagement.

Additional improvements to this work may be possible using the breadth of information available within well-documented and comprehensive corpora. For example, while the Blog Authorship Corpus included age and gender information about each blogger, this information was not used in the present study. To tailor prediction to individual users, it may be possible to limit the available vocabulary and gram-based statistics to information gathered from users of similar age and gender. This may improve prediction accuracy for both n-gram and sem-gram algorithms, as well as provide an approach to designing icon-based AAC devices that can evolve and adapt to users as their needs and abilities mature, potentially even suggesting new vocabulary words as the users age.

Acknowledgments

The authors would like to thank the creators of the Blog Authorship Corpus for making the corpus freely available for non-commercial research purposes. The material in this paper is based upon work supported by the National Science Foundation under Grant No. 0914808. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

J. Allan and H. Raghavan. 2002. Using part-of-speech patterns to reduce query ambiguity. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 307–314, New York, NY, USA. ACM.

D. Beukelman and P. Mirenda. 1998. *Augmentation and alternative communication: Management of severe communication disorders in children and adults*. Paul H. Brookes, Baltimore.

S. Bickel, P. Haider, and T. Scheffer. 2005. Predicting sentences using n-gram language models. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 193–200, Stroudsburg, PA, USA. Association for Computational Linguistics.

S. Bird, E. Klein, and E. Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, 1 edition, July.

A. C. Broerse and E. J. Zwaan. 1966. The information value of initial letters in the identification of words. *Journal of Verbal Learning and Verbal Behavior*, 5(5):441–446, October.

P. Demasco and K. McCoy. 1992. Generating text from compressed input: an intelligent interface for people with severe motor impairments. *Commun. ACM*, 35(5):68–78, May.

H. Fang and C. Zhai. 2006. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 115–122, New York, NY, USA. ACM.

R. Hemayati, W. Meng, and C. Yu. 2007. Semantic-based grouping of search engine results using WordNet. In *Proceedings of the joint 9th Asia-Pacific web and 8th international conference on web-age information management conference on Advances in data and web management*, APWeb/WAIM'07, pages 678–686, Berlin, Heidelberg. Springer-Verlag.

J. Higginbotham, A. Bisantz, M. Sunm, K. Adams, and F. Yik. 2009. The effect of context priming and task type on augmentative communication performance. *Augmentative and Alternative Communication*, 25(1):19–31.

Y. How and M. Kan. 2005. Optimizing predictive text entry for short message service on mobile phones. In *Human Computer Interfaces International (HCII 05)*.

A. Järvelin, A. Järvelin, and K. Järvelin. 2007. s-grams: Defining generalized n-grams for information retrieval. *Information Processing & Management*, 43(4):1005–1019, July.

G. Karberis and G. Kouroupetroglou. 2002. Transforming spontaneous telegraphic language to Well-Formed greek sentences for alternative and augmentative communication. In *Proceedings of the Second Hellenic Conference on AI: Methods and Applications of Artificial Intelligence*, SETN '02, pages 155–166, London, UK, UK. Springer-Verlag.

T. Kiss and J. Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Comput. Linguist.*, 32(4):485–525, December.

H. Koester and S. Levine. 1996. Effect of a word prediction feature on user performance. *Augmentative and Alternative Communication*, 12(3):155–168.

H. Koester and S. Levine. 1997. Keystroke-level models for user performance with word prediction. *Augmentative and Alternative Communication*, 13(4).

G. Lesher and G. Rinkus. 2002. Domain-Specific word prediction for augmentative communication. In *Proceedings of the RESNA 2002 Annual Conference*.

G. Lesher and C. Sanelli. 2000. A Web-Based system for autonomous text corpus generation. In *Proceedings of ISAAC*.

G. Lesher, B. Moulton, and J. Higginbotham. 1998. Techniques for augmenting scanning communication. *Augmentative and Alternative Communication*, 14(2):81–101, January.

J. Li and G. Hirst. 2005. Semantic knowledge in word completion. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, Assets '05, pages 121–128, New York, NY, USA. ACM.

C. Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.

Y. Lv and C. Zhai. 2009. Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 299–306, New York, NY, USA. ACM.

J. Matiasek and M. Baroni. 2003. Exploiting long distance collocational relations in predictive typing. In *Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods*, TextEntry '03, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mayer-Johnson. 2012. Picture communication symbols collections (http://www.mayer-johnson.com). March.

K. McCoy, C. Pennington, and A. Badman. 1998. Compansion: From research prototype to practical integration. *Natural Language Engineering*, 4(01):73–95.

S. Nikolova, M. Tremaine, and P. Cook. 2010. Click on bake to get cookies: guiding word-finding with semantic associations. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '10, pages 155–162, New York, NY, USA. ACM.

R. Patel, S. Pilato, and D. Roy. 2004. Beyond linear syntax: An Image-Oriented communication aid. *Journal of Assistive Technology Outcomes and Benefits*, (1):57–66.

M. F. Porter. 1997. Readings in information retrieval. chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Ranks. 2012. English stopwords (http://www.ranks.nl), March.

J. Schler, M. Koppel, S. Argamon, and J. Pennebaker. 2006. Effects of age and gender on blogging. In *Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*.

C. Suen. 1979. n-Gram statistics for natural language understanding and text processing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):164–172, April.

K. Trnka and K. McCoy. 2007. Corpus studies in word prediction. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, pages 195–202, New York, NY, USA. ACM.

K. Trnka, D. Yarrington, K. McCoy, and C. Pennington. 2006. Topic modeling in fringe word prediction for AAC. In *Proceedings of the 11th international conference on Intelligent user interfaces*, IUI '06, pages 276–278, New York, NY, USA. ACM.

E. Tzoukermann, J. Klavans, and C. Jacquemin. 1997. Effective use of natural language processing techniques for automatic conflation of multi-word terms: the role of derivational morphology, part of speech tagging, and shallow parsing. *SIGIR Forum*, 31(SI):148–155, July.

O. Udwin and W. Yule. 1990. Augmentative communication systems taught to cerebral palsied children - a longitudinal study:I. the acquisition of signs and symbols, and syntactic aspects of their use over time. *British Journal of Disorders of Communication*, 25(3):295–309, January.

H. Van Balkom and M. Welle Donker-Gimbrere. 1996. A psycholinguistic approach to graphic language use. *Augmentative and alternative communication: European Perspectives*, pages 153–170.

A. Van Den Bosch and P. Berck. 2009. Memory-based machine translation and language modeling. In *The Prague Bulletin of Mathematical Linguistics*.

A. Van Den Bosch. 2006. Scalable classification-based word prediction and confusible correction. *Traitement Automatique des Langues*, 46(2):39–63.

K. Vertanen and P. O. Kristensson. 2011. The imagination of crowds: Conversational AAC language modeling using crowdsourcing and large data sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 700–711. ACL.

T. Wandmacher and J. Antoine. 2006. Training language models without appropriate language resources: Experiments with an AAC system for disabled people. In *Proceedings of LREC*.

G. Ward. 2002. Moby word list: American english (http://www.gutenberg.org/ebooks/3201). Public domain in the USA.

S. J. Westerman and T. Cribbin. 2000. Mapping semantic information in virtual space: dimensions, variance and individual differences. *International Journal of Human-Computer Studies*, 53(5):765–787, November.

Widgit. 2012. About symbols (http://www.widgit.com), March.