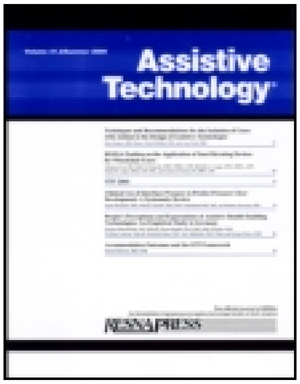

Assistive Technology: The Official Journal of RESNA
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uaty20

The Application of Natural Language Processing to Augmentative and Alternative Communication

D. Jeffery Higginbotham PhD [a], Gregory W. Lesher PhD [b], Bryan J. Moulton PhD [b] & Brian Roark PhD [c]

[a] Center for Excellence in Augmented Communication, Department of Communicative Disorders and Sciences, State University of New York at Buffalo

[b] DynaVox Technologies, LLC

[c] Center for Spoken Language Understanding, Oregon Health & Science University

Available online: 08 Mar 2012

To cite this article: D. Jeffery Higginbotham PhD, Gregory W. Lesher PhD, Bryan J. Moulton PhD & Brian Roark PhD (2012): The Application of Natural Language Processing to Augmentative and Alternative Communication, Assistive Technology: The Official Journal of RESNA, 24:1, 14-24

To link to this article: http://dx.doi.org/10.1080/10400435.2011.648714

Assistive Technology®, 24:14–24, 2012

ISSN: 1040-0435 print/1949-3614 online
DOI: 10.1080/10400435.2011.648714

Taylor & Francis
Taylor & Francis Group

The Application of Natural Language Processing to Augmentative and Alternative Communication

D. Jeffery Higginbotham, PhD,[1] Gregory W. Lesher, PhD,[2] Bryan J. Moulton, PhD,[2] and Brian Roark, PhD[3]
[1]Center for Excellence in Augmented Communication, Department of Communicative Disorders and Sciences, State University of New York at Buffalo
[2]DynaVox Technologies, LLC
[3]Center for Spoken Language Understanding, Oregon Health & Science University

ABSTRACT Significant progress has been made in the application of natural language processing (NLP) to augmentative and alternative communication (AAC), particularly in the areas of interface design and word prediction. This article will survey the current state-of-the-science of NLP in AAC and discuss its future applications for the development of next generation of AAC technology.

KEYWORDS natural language processing, NLP, augmentative and alternative communication, AAC, communication aids, word prediction

INTRODUCTION

Over the past three decades significant technical progress has been made in the augmentative and alternative communication (AAC) area as it concerns the processing of language materials for written and spoken communication. Early in the development of AAC technologies, researchers and manufactures utilized natural language materials for the development of word prediction and other rate and efficiency enhancing techniques (Baletsa, Foulds, and Crochetiere, 1976; Eulenberg, Reid, & Rahimi, 1977; Goodenough-Trepagnier, Tarry, & Prather, 1982). This early work, as well as the integration of natural language processing (NLP) research from other domains, has led to significant advances in the ways that spoken and written language is processed, presented, and used by those who rely on AAC for their communication. This article will survey the current state-of-the-science of NLP in AAC and discuss the future application in this area.

WHAT IS NATURAL LANGUAGE PROCESSING?

Suppose you received an email from a friend that read: "¿Quieres ir al cine a ver esa película nueva que Jenny dijo que parecía ser muy buena?" You realize immediately that this e-mail is not written in English. If you don't speak the language of the e-mail, you may want to know what the language is, want a literal translation, or maybe want just the gist of the e-mail (do you want to go to the movies?). The computational processes involved in this type

Address correspondence to D. Jeffery Higginbotham, 122 Cary Hall, Department of Communicative Disorders and Sciences, University at Buffalo, Buffalo, NY 14214. E-mail: cdsjeff@buffalo.edu

of task are collectively referred to as NLP, where the term "natural language" applies to the human language content of what is being processed by the computer. NLP is concerned with computer algorithms that analyze, modify, augment, or generate human language and methods that range from assigning probabilities to words or sequences of words (e.g., word prediction or completion) to full-scale transformation of sentences into new sentences (e.g., sentence simplification).

Among the many applications making core use of NLP algorithms are automatic machine translation from one language to another, extraction of structured information from large language corpora, speech recognition, and processing of spoken language. In AAC applications, NLP techniques have long been used in word prediction and optimized scanning overlays. Other areas such as dysarthric speech recognition, sentence simplification, context input, and brain computer interfaces are being actively researched (see Fager, Beukelman, Fried-Oken, & Jakobs, 2012 in this special issue).

Most NLP systems process input via statistical language models trained on observations of natural language using machine learning techniques. For example, word prediction and word completion models are often developed by collecting large text corpora (often exceeding 10 million words), then making predictions based on patterns observed in those collections. Similarly, language translation systems are frequently trained on large collections of translations. Resulting correspondences between phrases in one language and those of another are discovered in such collections and used to identify possible translations of new sentences. Other approaches may involve using a relatively small number of manually annotated examples, rules and/or patterns for processing natural language input. Because these approaches permit many possible outputs at any point in time, sometimes the results are perceived as being incorrect from the consumer's point of view. However, the overall accuracy that can be achieved is frequently sufficient to provide real utility.

A CASE STUDY IN NLP

To better understand what components are required and what methods are commonly used in an NLP application, a detailed example of the process of word completion and prediction will be provided. Word completion and prediction are commonly used in assistive keyboard software by providing a short list of predictable words for selection.

The most basic algorithm for word completion relies on a lexicon that can be searched for matching initial letters. For example, when typing "the pi..," the system can search the lexicon for words starting with the letters "pi" to retrieve and present candidates for words (e.g., <u>pi</u>ano, <u>pi</u>e, <u>pi</u>ece, <u>pi</u>stol, <u>pi</u>ty, etc.). Because the number of words in the lexicon that match the prefix will often exceed the number of prediction slots provided by the interface, there must be a method for deciding which words to present.

The first option might be to rank the candidates by their frequency of occurrence. If the individual has been using the system long enough, those statistics may be available from his or her performance log. In the absence of a large amount of individual specific observations, large text corpora are of high utility, since they allow for calculation of word or phrase frequency over a large sample of language. For the current example, the five most likely words prefixed by "pi" (e.g., say: pick, pitch, piece, picture, pitcher) would be placed in the word prediction region of the interface.

But there is more information than just the letter sequence "pi"–in our example, this prefix is preceded by the word "the." Statistical models that take into account some number of preceding words to determine the probability of each candidate word completion are known as n-gram models, where the "n" indicates the number of words that are examined. A 2-gram, or bigram model, uses the current letter sequence and previous word to predict the current word, a trigram model uses the current and previous two words, a quadgram model uses the current and previous three words, and so forth. These models capture extremely useful information about observed word sequences, are simple and efficient to use, and are widely employed for problems of this sort. Using the hypothetical bigram model, if we re-calculate our probabilities based on the preceding word "the," the five that become most likely now might include "pieces" and "pilot" instead of "pick" and "pitch."

Syntactic expectations in the form of part-of-speech tags or full statistical grammar models can also be used to determine the most likely continuations. For example, because "the" tends to be followed by nouns or noun modifiers, the probabilities of this subset of the total vocabulary can be adjusted appropriately. However, the additional computation and model complexity required to include such information often outweighs the benefit achieved in model quality, hence simple n-gram models of the sort described above are the norm.

To summarize: statistical language models (most often n-gram models), trained on language corpora and/or from prior user[1] system use, are used to rank words within a particular context and present them to the user as possible candidates. This application is quite typical of NLP systems in that it makes use of statistical models estimated from data to derive likely information of utility for the application. While the models and tasks differ, this data-driven paradigm is ubiquitous.

NLP APPLICATIONS IN AAC

Keyboards

From the early days of mechanical text entry, the statistics of language have been leveraged to improve input efficiency. The QWERTY keyboard layout, patented by Christopher Scholes in 1867, was a rough attempt to maximize typing speed by placing frequently co-occurring characters on opposite sides of the keyboard, thereby promoting alternating hand utilization. Later, August Dvorak established a set of explicit goals (e.g., the most common characters should appear in the home row and the most common digraphs should be the easiest to type) and used the statistical properties of English to optimize the key arrangement under these constraints.

Similarly, the optimization of single-selection text input has used statistical properties of written language to improve typing efficiency. For example, the FITALY[2] keyboard arrangement (Ichbiah, 1996) represents an early example in which the total distance traveled between selections within a 6×5 grid of keys is minimized by nearly 30% when compared to an alphabetic layout (Higginbotham & Lesher, 2004; Judge & Friday, 2011; Lesher, Moulton, & Higginbotham, 1998a).

Although keyboards with optimized layouts can theoretically improve text input rates and decrease fatigue by minimizing net motor activity, their unfamiliarity has prevented their widespread adoption (Baletsa et al., 1976; Lesher et al., 1998a). Most keyboards designed for direct selection—both within the AAC community and in the broader mobile market—still feature either alphabetic or QWERTY layouts. In the domain of scanning, however, the significant gains associated with optimizing the arrangement of the character grid often trump the familiarity of more traditional layouts. By rearranging a scanning grid from an alphabetic arrangement to one in which the most frequent characters appear in locations that are quicker to select, communication rate gains on the order of 30 to 40% can be achieved (Lesher et al., 1998b).

Various researchers have tried introducing an element of on-the-fly reorganization to scanning, either by supplementing a static grid with a short character prediction list or by dynamically rearranging all of the characters between each selection so as to ensure the minimum selection cost for the most probable letters. Despite the use of sophisticated n-gram models, it appears that the cognitive load imposed by these systems render them impractical except with the slowest of scanning rates (for a review of issues, see Lesher et al., 1998b).

Roark, de Villiers, Gibbons, and Fried-Oken (2010) have noted that even with dynamic reorganization, the structural constraints of traditional scanning systems (i.e., linear or row-column topologies) limit even their theoretical efficiency. They have proposed an alternative, called Huffman scanning that dynamically assigns optimal binary codes to each character after each selection. The characters are not rearranged dynamically in this paradigm, but rather they are highlighted in-place in correspondence with their binary codes. Even though the cognitive loads associated with Huffman scanning are higher than for traditional systems, the improvement in efficiency (combined with a built-in method for seamless error correction) has led to faster text output rates and much lower error rates in preliminary studies.

Given an input environment in which it is not possible to accurately choose one of 30 or more keys via direct selection, one solution is to reduce the number of keys by placing multiple characters on each key. As the user selects each key, ambiguity is resolved by examining the relative probabilities of the various possible combinations of ambiguous selections. For example, if a user were to enter the telephone keypad sequence [pqrs] [abc] [ghi] [def], he or she might have intended "rage," but it's more likely that "said" was the real goal, especially if the preceding two words were, for example, "what she." The n-gram techniques described previously are generally used to establish the word probabilities used for disambiguation.

Although ambiguous keyboards of this type gained widespread popularity with the appearance of the T9 text entry system for cell phones, they have a rich history within the context of AAC (Kushler, 1998; Lesher et al., 1998a; Lesher & Moulton, 2000; Levine, Trepagnier, Getschow, & Minneman, 1987). It's possible to improve the efficiency of such keyboards by grouping characters so as to minimize statistical ambiguity, but as with FITALY-like attempts to optimize conventional keyboard arrangements, the public has generally chosen to instead use the more familiar traditional groupings found on telephone keypads.

For text entry on mobile device touchscreens, gesture-based keyboard input has become popular with the recent appearance of commercial entries such as Swype and ShapeWriter. Interestingly, Nantais, Shein, and Treviranus (1994) had long ago proposed a similar method for AAC. In keyboard gesture systems, the user spells out the desired word on an onscreen keyboard via a continuous gesture through the letters of the word rather than by discrete taps on individual keys. Although a precisely executed gesture that passes through the heart of each target letter may uniquely identify a particular word (in a given language), there are many words for which the gestures are indistinguishable (for example, the gestures for "pit" and "pot" on a QWERTY keyboard both consist of a straight line between "p" and "t"). More importantly, since input speed is a primary consideration, gestures are not meant to be precisely executed. In practice, nearly every gesture will contain significant ambiguity. The probability that a gesture corresponds to a given word can be computed by correlating the gesture with the locations of the component letters of that word, taking into account the NLP likelihood of the word within the current context, the mechanics of the gesture (for example, the tendency to overshoot), and the past history of user gestures. As described above for conventional and ambiguous keyboards, it's also possible to improve efficiency for keyboard gesture input by rearranging the keys—primarily by statistically minimizing gesture ambiguity (as in the ATOMIK[3] layout from ShapeWriter; Kristensson & Zhai, 2007; Zhai et al., 2009).

Although keyboard gesture systems have proven popular on mainstream devices, it remains to be seen how useful they will be for augmentative communication. One suspects that the execution of smooth, continuous motions through entire words will be difficult for a significant population. However, Beukelman and colleagues have been exploring the use of gestures through the first two or three letters in a word as a method of priming an n-gram word prediction system with the prefix context of the current word, with intriguing results (Beukelman, Schley, Ternus, & Fager, 2010; Fager, Beukelman, & Jakobs, 2010). Regardless of the utility of gestures themselves, the idea that imprecise input can be corrected or disambiguated using NLP can be applied to nearly any form of direct input. Many of the tap-based onscreen keyboards for mobile devices (such as SwiftKey[4] and the iPhone keyboard) now incorporate at least some degree of "sloppy input" disambiguation. Obviously, this would be an appealing feature for persons with impaired motor functions.

We must stretch the definition of keyboard somewhat to accommodate one final entry in this section. In the Dasher system (Ward, Blackwell, & MacKay, 2000), characters are presented to the user in a flowing, alphabetically-ordered spatial stream in which the ease of selecting a particular letter is correlated with its probability within the current context. The user controls the flow of the stream using vertical mouse movement to change the target character and horizontal mouse movement to control the rate of the flow—a dynamic that is difficult to explain, but relatively easy to demonstrate.[5] The advantage of Dasher over other dynamically arranged keyboards is that the overall alphabetic "layout" of Dasher is never changed, only the proportional area subsumed by each character. Additionally, the user is provided with advance information about how to select future characters since they are represented downstream from the current set of characters—a kind of preview of what's coming. Recognizing that precise 2D mouse control may be impossible for many potential Dasher users, MacKay and his colleagues at Cambridge University have suggested an impressive array of alternative input methods, including systems designed for use with single-dimensional sliders, one or two switches, and eye-trackers.

Word Prediction

As perhaps the highest profile application of NLP in augmentative communication, word prediction exists in some form on virtually all high-tech communication aids. In one early incarnation (Eulenberg et al., 1977), a frequency-tagged word list was used to suggest words for immediate direct selection. This was followed by more sophisticated systems using the n-gram methods outlined above (Swiffin, Arnott, Pickering, & Newell, 1987). Nearly all commercial AAC prediction engines now use some form of n-gram prediction. The efficacy of n-gram word prediction in reducing the number of selections necessary to produce a given text (i.e., keystroke savings) depends on the number of words in the prediction list, the order of the n-gram, the method of blending n-gram components, the size and nature of the source corpus, and the nature of the text being produced. As a general rule, however, with a 6-item prediction list, keystroke savings from commercial AAC systems fall in the 40% to 50% range (from Higginbotham, 1992, despite the 20 years of elapsed time, still an accurate assessment). It is clear, however, that these values fall short of the theoretical limits, which some researchers put somewhere between 60% and 65% (Copestake, 1997; Lesher, Moulton, Higginbotham, and Alsofrom, 2002). A brief overview of the use of word prediction in AAC will be provided here; readers more interested in this topic are encouraged to consult the excellent survey of Garay-Vitoria and Abascal (2006).

The efficacy of word prediction in improving communication rates has been an issue of some contention. If keystroke savings alone determined communication rate, one would expect a 50% savings to result in a 100% increase in communication rate. Given the cognitive loads associated with searching the list and the dynamic planning associated with targeting a word prediction key, however, it was generally accepted that word prediction actually slows communication for both direct selection (Koester and Levine, 1996) and scanning (Koester and Levine, 1994) for most users. However, recent work by Trnka, Yarrington, McCaw, and McCoy (2009) suggests that with more advanced word prediction algorithms, significant communication rate gains may be achieved. Note that other factors, such as fatigue reduction, usability and task demands may play significant roles in determining whether to use word prediction (e.g., Higginbotham, Bisantz, Sunm, Adams, & Yik, 2009).

Much of the cutting edge word prediction research of the past few years has focused on sophisticated prediction engines capable of supporting multiple languages. Although some of the advanced techniques utilized in these systems are designed to explicitly address the characteristics of European languages other than English (e.g., the fact that they are much more highly inflected), these methods also can improve performance in English. The impressive SIBYLLE system (Wandmacher, Antoine, Poirier, & Départe, 2008) combines quadgram prediction with a sophisticated latent semantic analysis (LSA) component to yield baseline keystroke savings of nearly 58% for French. The FASTY design incorporates n-grams derived from both words and parts of speech, modulated by a rule-based grammar engine to provide predictions in German, Dutch, Swedish, and French (Matiasek, Baroni, and Trost, 2002). Working on an English prediction system, Li and Hirst (2005) have selectively biased n-gram-predicted words that are semantically related to recently appearing words. Combined with a novel method to deal with proper nouns, they report an improvement of an astonishing six percentage points in keystroke savings over a vanilla n-gram model.

A recent focus of word prediction research has been the use of domain-specific predictions to boost keystroke savings, where the domain might be determined by conversational topic, geographic location, time of day, or a host of other contextual factors. The most obvious domain is the user's daily language productions. Trnka, Yarrington, McCoy, and Pennington (2006) have shown that blending topic-specific predictions with those derived from a general purpose corpora can provide significant gains in keystroke savings. Recent studies of genre-specific prediction have underscored the need to properly match the domain used for n-gram training to the type of text actually being generated by users (Wandmacher et al., 2008).

Text input systems that exploit character probabilities, such as dynamically rearranged scanning layouts, have historically used n-gram character models to estimate these probabilities, where n typically falls in the range of three through eight. By definition, such an approach limits the amount of context that can be utilized by the system. However, if an effective word prediction engine is available, it is straightforward to adapt it to produce character likelihoods by tabulating the probabilities of each character summed across all predicted words. When blended with traditional n-gram character predictions to smooth the data, this technique can provide significantly more accurate estimates of character probabilities (Lesher & Rinkus, 2002). If word prediction is n-gram based, this approach is roughly analogous to using n-gram character prediction with large values of n. However, word-based character prediction lends itself to any type of word prediction engine, including non-n-gram methods, opening up new possibilities for accurate character prediction.

Information generated by word prediction engines can be leveraged for many purposes other than traditional word prediction. For example, word probabilities can be used to bias the choices offered by spell checkers, or to automatically correct minor spelling errors outright. Prediction engines that utilize syntactic or semantic components can similarly used for grammar correction and for sense disambiguation. Widgit's Communicate software[6] uses syntactic disambiguation to provide the appropriate symbol when an ambiguous word like "can" appears in a sentence. In general, probability information generated by an effective prediction engine can be utilized in many settings in which there is some textual ambiguity.

Speech Recognition

Automatic speech recognition (ASR) takes human speech as input, and outputs the text of the words that were spoken. Falling within the broader range of human language technologies, ASR makes use of NLP methods such as statistical language models, but because it requires digital signal processing of the acoustic speech signal, it is typically treated separately from written language processing. Because of the difficulties inherent in the machine recognition of natural speech, early commercial ASR products, such as dictation software, would typically require the individual to engage in a training regime, such as reading from a predefined passage, in order to provide the system with enough input for it to learn the idiosyncrasies of the individual's speech. As with other NLP applications, advances over the past decades now enable individuals without any personalized training to be able to talk over the phone to customer service applications or perform a voice search. To the extent that an individual's speech patterns are atypical—due to a regional or non-native accent or to a speech disorder—such general-use technologies will have less utility and will require specialized solutions.

In very broad strokes, speech recognition systems work by using two kinds of statistical models: an acoustic model that measures the goodness of fit between the acoustics and the candidate word sequence, and a language model that measures the goodness of fit between the candidate word sequence and the particular language being spoken. For example, if a speaker says, "I'm okay" the recognizer will consider various possible transcriptions, which may include the actual intended words (hopefully), as well as things like "I'm Mo Kay" and "Eye no gate." Depending on the speaker, all of these may have relatively high probabilities according to the acoustic model (i.e., they are relatively good fits to the models of how words are pronounced). The language model serves to disambiguate between candidates that are acoustically plausible, preferring those which are similar to what has been "observed" in that language before. Those observations typically come from large corpora, and the models are trained to recognize a closed vocabulary—a word has to be explicitly included in the vocabulary in order to be recognized. Typical vocabulary sizes in English will be more than 10,000 words.

If speech is unimpaired, commercially available automatic speech recognition systems can be used to perform dictation. As mentioned above, the system can be tuned to the individual—typically through reading of a short text—to achieve better performance than is possible without tuning. Most products will also allow personalization of the language model by uploading documents of various sorts, to provide names, words and phrases that the individual will likely be using. For speakers with dysarthria, however, such approaches will prove less accurate, hence with diminished utility as an alternative communication approach (Beukelman et al., 2007; Fager, Beukelman, Jakobs, & Hosom, 2010). Research continues on improving speech recognition for dysarthric speech, but for practical use, ASR for these individuals appears some time off in the future.

Within a multi-modal interface, however, dysarthric speech can provide additional information that can yield superior AAC keystroke savings. The Speech Supplemented Word Prediction Program (Fager et al., 2010; Hosom, Jakobs, Baker, & Fager, 2010) is an approach that makes use of low-intelligibility speech along with letter selection to yield improved word prediction.[7] The system user types the first letter of the intended word and speaks the word. The system then performs automatic recognition of the spoken word, constrained by the initial letter, then returns a predicted word, which can be selected by the user, if correct or otherwise ignored. Significant additional keystroke savings can be achieved with this system over typical word prediction methods, even for subjects with relatively low speech intelligibility levels (Fager et al., 2010).

Processing the Context

So far, discussion has focused on processing linguistic information in the forms of text and speech, in order to improve expressive communication in AAC. However, advances in geopositioning, image processing, Internet availability, and computing capability offer a wide range of possibilities of intelligently including different aspects of one's immediate environment and social context into the content of the AAC device. The possibility of priming a word predictor or otherwise optimizing an AAC system with information derived from knowledge about the communication partner, discourse genre, conversational topic, user's location, and time has intrigued AAC researchers and developers. The potential of these resources are just being realized in the research and commercial arenas.

Mining the linguistic context

Using information about the discourse genre and conversational topic is not only intuitively attractive, but has been actively researched within corpus linguistics (Biber, Conrad, & Reppen, 1991, 1998) and NLP (Blei, Ng, & Jordan, 2003; Blitzer, McDonald, & Pereira, 2006). A variety of discourse genres (e.g., narrative, expository, procedural description) can be identified and differentiated from one another, based on a variety of syntactic, grammatical and morphological characteristics. In addition, topical talk can be characterized by distinct constellations of semantic information. AAC research in word prediction has also investigated the impact of discourse genre and topic on prediction efficiency.

Several investigations have examined the potential of topic priming, providing a word predictor with topic relevant texts or transcripts (Higginbotham et al., 2009; Lesher & Rinkus, 2002; Trnka, 2008; Trnka et al., 2006; Trnka et al., 2009). Topic-informed predictors have been shown to display keystroke savings improvements over base systems, depending on dictionary size, types of topical materials, and experimental context (machine-simulation, human-transcription, human-interaction) (Higginbotham, 1992; Higginbotham, Lesher, & Luo, 2008; Lesher et al., 2002; Wandmacher & Antoine, 2007).

Utilizing web-based information

A basic problem with most topic priming techniques is the inability of the prediction system to deal with topic materials that aren't pre-programmed into the system. The ability to discuss current events, politics, entertainment, sports, and so forth, often eludes augmented speakers because a topic-specific vocabulary system is not available on their devices. Current approaches involve uploading or programming topic vocabulary, which is non-optimal, both in terms of the time and effort required to input the vocabulary, and that the vocabulary is not fully integrated into the device's language model.

One solution is to search and retrieve topical language material from the Internet whenever a new topic arises. Researchers at the University at Buffalo and DynaVox Technologies have developed an AAC approach that integrates the results of Internet topic searches (both spontaneous and pre-programmed) directly into the user's word prediction system. In initial testing, topic searches with the experimental Webcrawler produced a 53% keystroke savings for topic-specific written texts, 5 percentage points above the already robust trigram-based word predictor used in the AAC system (Higginbotham et al., 2008). Field-testing has resulted in further refinements of the system and a commercial release planned for the future (Fulcher, 2011).

Partner recognition

Recognition of the speaking partner by the AAC device could provide a wealth of information including a shared conversation history (topics, vocabulary), mutual interests and personal and role-related relationships. In their work with locked-in patients, Davis, Moore, and Story (2003) provide a knowledge framework for developing a partner recognition system. They propose describing each partner according to personal characteristics, level and knowledge of topics, perceptual and motor skills, conversation history and mood (emotions expressed during last conversation), then matching them with the user's own profile to provide context relevant talk. Technologies for recognizing faces and voices are now becoming commercially available; however, to date, no study has empirically demonstrated the usefulness of partner identification in AAC.

Processing partner talk

Rather than attempting to recognize the partner's identity, it may be possible to use the current talk of one's interlocutor to inform the AAC device. Research at the University at Buffalo has been examined the potential of partner talk for improving word prediction.

Wisenburn and Higginbotham (2008, 2009) developed an AAC prototype (Converser) that identified the noun phrases spoken by the communication partner (via speech recognition) and incorporated them into the AAC device. Modest gains in communication rate and user preference were found for Converser compared to an alphabet typing system, despite the fact that the speech recognition accuracy was low.

Min (2012) explored the effects of partner talk by analyzing extended transcriptions of multiple participants engaged in a variety of social interactions. By inputting transcripts of spoken conversations into a word prediction emulator, the contributions of communication partner talk were studied in terms of its ability to improve word prediction. Preliminary results support previous research on corpora of natural speakers showing modest increases in keystroke savings depending on discourse genre and word predictor variables (e.g., sophistication of the prediction algorithm). Related work at the University at Buffalo is focused on leveraging partner talk to optimize the predictability of fringe vocabulary and for triggering semantically related words.

Text simplification and summarization

Another NLP technology with significant potential, but yet unrealized application for AAC, concerns the automatic reconfiguration of existing text. In text simplification, documents are transformed into simpler sentence structures and vocabulary while retaining the meaning of the original. For example, the original document would be segmented into individual clauses, and reconstituted into multiple sentences that are easier to understand (Original: "The plane, a twin-engined Cessna owned by XYZ, Inc., crashed into the ocean after striking a flock of seagulls on take-off." Simplified: "The plane crashed into the ocean. The plane was a twin-engined Cessna. The plane was owned by XYZ, Inc. The plane hit birds during take-off.").

A related NLP technology, automatic summarization, produces short summaries of documents or collections of documents. Similar to text simplification, the aim is to produce a short, simple overview of what is presented in a document. Automatic summarization systems work by finding sentences in the document that are particularly important, and then extracting the most important sentences to include in the summary. Returning to the previous example, the automatic summary might only include two of the four simplified clauses: "The plane crashed into the ocean. The plane hit birds during take-off."

Although not currently employed in any commercial AAC technologies, the potential for restructuring text obtained from the Internet, e-books, e-mail, transcribed partner speech, and so forth, is significant. Complex reading materials could be simplified for individuals with cognitive-linguistic challenges for palatable reading and/or listening experiences. Newspaper stories, magazine articles, web page content, e-mails, and so forth, could be processed and inserted into the individual's AAC system, transformed into materials for expressive communication. Provided with a simplified and appropriately segmented set of topic materials, the augmented speaker could select from these offerings to discuss current events, baseball scores or other topics related to their personal lives.

Location

Knowledge about one's location can be used to provide important information related to one's whereabouts or context specific activities. This might include location-relevant words and utterances, page organizations, maps, locations of favorite places, historical content from conversations that occurred at the same location or in similar establishments (e.g, Starbucks, clothing store). This information is potentially available by coordinating one's location (e.g., via GPS) with web-based information sources (e.g., Google maps) and a geographic-tagged language corpus in one's device. Although the need and interest for location specific vocabulary has been demonstrated, substantial challenges remain in utilizing location information for AAC (Bryen, 2010; DeRuyter, McNaughton, Caves, Bryen, & Williams, 2007).

Patel and colleagues (Dominowska, Roy, & Patel, 2002; Patel & Radhakrishnan, 2007) analyzed a geographically-tagged corpora of 6,300 utterances (60,794 words) and organized the vocabulary into 8 different location-specific topic areas (e.g., class, kitchen, lab meeting, grocery store). Using a variety of data mining strategies, they were able to organize vocabularies in terms of frequency and semantic relatedness and develop a proof-of-concept prototype with their IconChat AAC software. Recently, commercial developers have introduced mobile applications that provide an interface between the mobile device, GPS and the kinds of information about nearby businesses that can be used for communication purposes (e.g, MyVoice[8]; Locabulary[9]). While providing the user with location-relevant words and phrases, these applications do not make use of NLP for vocabulary search, retrieval or display.

Several challenges to location-enhanced AAC technology remain. First, how will location-relevant vocabulary be identified and made available? To date, the solution has been to provide a tailored set of vocabulary items. Although a necessary first step, hand-tailored static vocabularies may be insufficient to address new and/or unanticipated places, events, or circumstances encountered by the augmented speaker. Because of the small size of the AAC industry, it is unlikely that developers will be able to produce tailored vocabularies that will be sufficiently specific and robust to meet new and/or unanticipated situations (Bryen, 2010). Relying on the volunteers from the AAC user-community to provide context-specific vocabulary sets is an intriguing idea, but possesses the same problems discussed above. One solution may be to augment tailored vocabularies using a web crawler-type system, which would search for location-specific vocabulary or query a crowd-sourced feedback system in which utterances are compiled from other users' communications collected in similar locations and stored in the cloud. These strategies could be further augmented by structuring language materials with relevant utterance frames using an NLP-style inference engine that determines the current communication genre (e.g., transactional versus interpersonal), then provides a set of utterances to carry out the actions associated with the particular situation.

Time

Time (i.e., clock time, event, schedule, calendar, seasons) could also be employed as contextual input for NLP processing. The regularity of a daily schedule (e.g., getting dressed, meals, school/work) could be used as input to prime a word predictor or provide content to a communication page with vocabulary related to time of day (e.g., provide breakfast options on a meal page). Further benefits could be achieved by combining time with other contextual information sources, like location (e.g., at home versus at a restaurant). External scheduling information, such as meetings or events (e.g., football game or favorite television show), could also be used to derive the regularities of talk associated with that event. Information related to the seasons, upcoming holidays and other specific dates could provide similar benefits by making language related to those repeated occasions available for speaking. Recent work by Reiter et al. (2009) has explored the processing of student schedules with information about their actual experiences to facilitate the production of personal narratives.

FUTURE DIRECTIONS

From the above discussion it should be apparent that the NLP techniques are ubiquitous with current AAC technologies and will continue to provide important resources for future development. One would expect that the most dramatic NLP-influenced development will involve integrating information about the individuals who rely on AAC context into the device. It is not too hard to imagine future AAC systems that provide the resources (e.g., recordings of an individual's daily experience, GPS, internet searches), and the language and discourse support (e.g., context-based word prediction, context-sensitive syntactic frames) that actively support extemporaneous story telling during conversation. As noted, most of the information sources are already available and now it is up to the manufacturers to integrate them into their AAC technologies.

It is likely that the future development of basic NLP processes will incremental. Their future contribution to AAC will likely be through broadening the application of NLP into areas like spell-checking, context-based thesaurus, and genre-based word prediction. As evident in this review, the current research and development focus in on the NLP technology itself–improving processing performance and applying NLP to new problems. Issues of how NLP is used by different populations within the AAC community, its integration and use across different interfaces and graphic representation systems, and its ability to be adapted to enhance communication across a variety of activities remain topics that need to be explored in the near future.

NOTES

1. Although the term "user" is not a preferred term in the disability community it is used widely in the literature dealing with NLP applications and user-interface design. In this paper this term will be refer to the "generic" person interacting with technology.
2. http://en.wikipedia.org/wiki/FITALY
3. http://www.almaden.ibm.com/u/zhai/ATOMIK.htm
4. http://www.swiftkey.net/
5. http://www.inference.phy.cam.ac.uk/dasher/
6. http://www.widgit.com/products/symwriter/index.htm
7. A video of the SSR system can be found here: http://www.youtube.com/davinciawards#p/c/19254BCD97F0FFF8/0/bTks5CPM8Ks
8. http://myvoiceaac.com
9. http://locabulary.com

REFERENCES

Baletsa, G., Foulds, R., & Crochetiere, W. (1976). Design parameters of an intelligent communication device. In *Proceedings of the 29th Annual Conference on Engineering in Medicine and Biology*. Chevy Chase, MD: Alliance for Engineering in Medicine and Biology.

Beukelman, D. R., Fager, S., Ball, L., & Dietz, A. (2007). AAC for adults with acquired neurological conditions: A review. *Augmentative and Alternative Communication*, *23*(3), 230–242.

Beukelman, D., Schley, H., Ternus, T., & Fager, S. (2010). New AAC access strategy: Gesture enhanced word predication. *American Speech, Language, Hearing Association, 2010 Convention*, Philadelphia, PA.

Biber, D. (1991). *Variation across speech and writing*. Cambridge: Cambridge University Press.

Biber, D., Conrad, S., & Reppen, R. (1998). *Corpus linguistics: Investigating language structure and use*. Cambridge: Cambridge University Press.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Blitzer, J., McDonald, R., & Pereira F. (2006). Domain adaptation with structural correspondence learning. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 120–128.

Bryen, D. N. (2010). Communication during times of natural or man-made emergencies: The potential of speech-generating devices. *International Journal of Emergency Management*, *7*(1), 17–27.

Copestake, A. (1997). Augmented and alternative NLP techniques for augmentative and alternative communication. In *Proceedings of the ACL Workshop on Natural Language Processing for Communication Aids*. Madrid.

Davis, A. B., Moore, M. M., & Storey, V. C. (2003). Context-Aware communication for severely disabled users. *ACM SIGCAPH Computers and the Physically Handicapped*, *73–74*, 106–111.

DeRuyter, F., McNaughton, D., Caves, K., Bryen, D. N., & Williams, M. B. (2007). Enhancing AAC connections with the world. *Augmentative and Alternative Communication*, *23*(3), 258–270.

Dominowska, E., Roy, D., & Patel, R. (2002). An adaptive context-sensitive communication aid. In *Proceedings of the 17th Annual International Conference on Technology and Persons with Disabilities*.

Eulenberg, J. M., Reid, R. J., & Rahimi, M. A. (1977). Representation of language space in speech prostheses. *In Proceedings of the 4th Conference on Systems and Devices for the Disabled*, Seattle, WA.

Fager, S. K., Beukelman D. R., & Jakobs, T. (2010). New AAC access strategy—Gesture enhanced word prediction. Presented, *American Speech Language and Hearing Association Annual Conference*, Philadelphia, PA.

Fager, S. K., Beukelman, D. R., Jakobs, T., & Hosom, J. P. (2010). Evaluation of a speech recognition prototype for speakers with moderate and severe dysarthria: A preliminary report. *Augmentative and Alternative Communication*, *26*(4), 267–277.

Fulcher, K. (2011). Webcrawler progress report. Unpublished report. University at Buffalo.

Garay-Vitoria, N., & Abascal, J. (2006). Text prediction systems: A survey. *Universal Access in the Information Society*, *4*(3), 188–203.

Goodenough-Trepagnier, C., Tarry, E., & Prather, P. (1982). Derivation of an efficient nonvocal communication system. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *24*(2), 163–172.

Higginbotham, D. J. (1992). Evaluation of keystroke savings across five assistive communication technologies. *Augmentative and Alternative Communication*, *8*(4), 258–272.

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

Higginbotham, D. J., & Lesher, G. W. (2004). The ambiguous keyboard. *Perspectives on Augmentative and Alternative Communication*, April, 12–16.

Higginbotham, D. J., Lesher, G. W., & Luo, F. (2008). Using a web crawler to enhance AAC. *International Society for Augmentative and Alternative Communication*, Montreal, Canada. ISAAC.

Hosom, J. P., Jakobs, T., Baker, A., & Fager, S. (2010). Automatic speech recognition for assistive writing in speech supplemented word prediction. In *Eleventh Annual Conference of the International Speech Communication Association*, Makuhari, Japan.

Ichbiah, J. D. (1996). Method for designing an ergonomic one-finger keyboard and apparatus therefore. *U.S. Patent #No. 5,487,616*. Washington, DC: U.S. Patents and Trademarks Office.

Judge, S., & Friday, M. (2011). Ambiguous keyboard for AAC. *Journal of Assistive Technologies*, *5*(4), 249–256.

Koester, H. H., & Levine, S. P. (1994). Learning and performance of able-bodied individuals using scanning systems with and without word prediction. *Assistive Technology*, *6*(1), 42–53.

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

Kristensson, P. O., & Zhai, S. (2007). Command strokes with and without preview: Using pen gestures on keyboard for command selection. In CHI '07: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Los Angeles.

Kushler, C. K. (1998, March). AAC: Using a reduced keyboard. *CSUN '98: California State University Conference on Disability Technology*, Los Angeles.

Lesher, G. W., & Moulton, B. J. (2000). A method for optimizing single-finger keyboards. In *Proceedings of the RESNA 2000 Annual Conference*, Orlando.

Lesher, G. W., Moulton, B. J., & Higginbotham, D. J. (1998a). Optimal character arrangements for ambiguous keyboards. *IEEE Transactions on Rehabilitation Engineering*, *6*(4), 415–423.

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

Lesher, G. W., Moulton, B. J., Higginbotham, D. J., & Alsofrom, B. (2002). Limits of human word prediction performance. *Proceedings of the California State University Northridge Conference on Disability Technology*, Los Angeles.

Lesher, G. W., & Rinkus, G. J. (2002). Leveraging word prediction to improve character prediction in a scanning configuration. In *Proceedings of the RESNA 2002 Annual Conference*.

Levine, S. H., Trepagnier, G. C., Getschow, C. O., & Minneman, S. L. (1987). Multi-character key text entry using computer disambiguation. In *RESNA 10th annual conference*, San Jose.

Li, J., & Hirst, G. (2005). Semantic knowledge in word completion. In *Proceedings of the 7th international ACM SIGACCESS Conference on Computers and Accessibility*, ACM: New York.

Matiasek, J., Baroni, M., & Trost, H. (2002). FASTY—A multi-lingual approach to text prediction. In *ICCHP '02: Proceedings of the Computers Helping People with Special Needs Conference*. Linz, Austria: Springer.

Min, H. (2012). Exploring the contribution of conversational context in word prediction. Unpublished manuscript, State University of New York at Buffalo.

Nantais, T., Shein, F., & Treviranus, J. (1994). A predictive selection technique for single-digit typing with a visual keyboard. *IEEE Transactions on Rehabilitation Engineering*, *2*(3), 130–136.

Patel, R., & Radhakrishnan, R. (2007). Enhancing access to situational vocabulary by leveraging geographic context. *Assistive Technology Outcomes and Benefits*, *4*(1), 99–114.

Reiter, E., Turner, R., Alm, N., Black, R., Dempster, M., & Waller, A. (2009). Using NLG to help language-impaired users tell stories and participate in social dialogues. In *Proceedings of the 12th European Workshop on Natural Language Generation*, Association for Computational Linguistics: Stroudsburg.

Roark, B., de Villiers, J., Gibbons, C., & Fried-Oken, M. (2010). Scanning methods and language modeling for binary switch typing. In *Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*, Association for Computational Linguistics: Stroudsburg, PA.

Shannon, C. E. (1951). Prediction and entropy of printed english. *Bell Systems Technical Journal*, *30*, 30–64.

Swiffin, A. L., Arnott, J. L., Pickering, J. A. & Newell, A. F. (1987). The use of syntax in a predictive communication aid for the physically impaired. In *Proceedings of the Tenth Annual Conference on Rehabilitation Technology*: San Jose, CA.

Trnka, K. (2008). Adapting word prediction to subject matter without topic-labeled data. In *Proceedings of the 10th international ACM SIGACCESS Conference on Computers and Accessibility*, ACM: New York, NY.

Trnka, K., Yarrington, D., McCaw, J., & McCoy, K. (2009). User interaction with word prediction: The effects of prediction quality. *ACM Transactions on Accessible Computing*, *1*(3), 1–34.

Trnka, K., Yarrington, D., McCoy, K., & Pennington, C. (2006). Topic modeling in fringe word prediction for AAC. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*. ACM: New York, NY.

Wandmacher, T., & Antoine, J. Y. (2007). Methods to integrate a language model with semantic information for a word prediction component. In *EMNLP '07: Proceedings of the conference on empirical methods in natural language processing*: Prague.

Wandmacher, T., Antoine, J. Y., Poirier, F., & Départe, J. P. (2008). Sibylle, an assistive communication system adapting to the context and its user. *ACM Transactions on Accessible Computing*, *1*(1), 1–30.

Ward, D. J., Blackwell, A. F., & MacKay, D. J. C. (2000). Dasher—a data entry interface using continuous gestures and language models. *In Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, ACM: New York, NY.

Wisenburn, B., & Higginbotham, D. (2008). An AAC application using speaking partner speech recognition to automatically produce contextually relevant utterances: Objective results. *Augmentative and Alternative Communication*, *24*(2), 100–109.

Wisenburn, B., & Higginbotham, D. (2009). Participant evaluations of rate and communication efficacy of an AAC application using natural language processing. *Augmentative and Alternative Communication*, *25*(2), 78–89.

Zhai, S., Kristensson, P. O., Gong, P., Greiner, M., Peng, S. A., Liu, L. M., & Dunnigan, A. (2009). Shapewriter on the iphone: From the laboratory to the real world. In *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems*, ACM: New York, NY.