An NLP-based Reading Tool for Aiding Non-native English Readers

Mahmoud Azab Ahmed Salama Kemal Oflazer
Carnegie Mellon University-Qatar
Doha, Qatar
{mazab,ahmedsaa,ko}@qatar.cmu.edu

Hideki Shima Jun Araki Teruko Mitamura
Carnegie Mellon University,
Pittsburgh, PA, USA
{hideki,junaraki,teruko}@cs.cmu.edu

Abstract

This paper describes a text-reading tool that makes extensive use of widely-available NLP tools and resources to aid non-native English speakers overcome language related hindrances while reading a text. It is a web-based tool, that can be accessed from browsers running on PCs or tablets, and provides the reader with an intelligent e-book functionality.

1 Introduction and Motivation

In this paper, we describe our approach in building a NLP-powered tool to aid in reading texts in English by non-native readers of the language, especially in an educational setting. Text, being bland, is hardly a conducive and motivating medium for learning, especially when the reader does not have access to aids that would enable her to get over minor and not-so-minor roadblocks ranging from unknown vocabulary to unrecognized and forgotten names, hard-to-understand sentences, issues with the grammar and lack of or forgetting the prior context in a former session of reading. We aim to make reading an active and interactive experience by enabling the user to interact with the text in a variety of ways using anytime-anywhere contextually guided access to textual information.

Our system is based on significant preprocessing and annotation of a library of texts using many publicly available NLP components for English, integrated in a UIMA (Unstructured Information Management Architecture) based server (Ferrucci and Lally, 2004). These annotated documents are then accessed via browser-based clients which essentially look like traditional e-book reading environments but with a much richer set of user accessible functionality. Thus our system can also be seen as a *showcase application for demonstrating English NLP tools and resources*. Our contribution is the integration of many publicly available tools and resources for English into a large-scale usable application implemented in a client-server software architecture structured around UIMA, along with work on development of some annotation components and/or combination of available ones.

In the rest of this paper, after a brief review of the use of NLP to help for reading, we will elaborate on the user visible functionality of our system and then present the software architecture and the implementation. Our system has been implemented save for a couple of features and we are now in the process of planning an intrinsic evaluation followed by a deployment to have it be used to gauge if student users find it effective.

2 Using NLP in Reading Aids

Recently, Computer Assisted Language Learning (CALL) systems have started making use of advanced language technology to build intelligent systems to aid and assess reading comprehension. An early project, GLOSSER Project (Nerbonne et al., 1997) developed a system that aids readers of foreign language text, by providing access to a dictionary, exploiting morphological analysis and part-of-speech disambiguation. The FreeText Project (Hamel and Girard, 2000), developed a NLP-based CALL system for intermediate to advanced learners of French. The LISTEN project at CMU on the other hand, has aimed to tutor elementary school students in reading English text by using speech technology (Mostow and Aist, 2001).

The REAP (Reader Specific Lexical Practice) project (Heilman et al., 2006), aimed at selecting individualized practice reading documents from the web using lexical, syntactic and readability levels. REAP chooses documents that contain cer-

Proceedings of Recent Advances in Natural Language Processing, pages 41–48,
Hissar, Bulgaria, 7-13 September 2013.

tain target vocabulary words that a student needs to learn. It also presents the documents within a web browser-based application along with a dictionary to provide word meanings and a set of automatically generated set of closed questions as an exercise. Recently, Eom et al. (2012) presented a system that incorporates word sense disambiguation for vocabulary assistance. Maamouri et al. (2012) presents, ARET (Arabic Reading Enhancement Tool) that aids the readers of Arabic as a second language. It provides the user with the morphological analyses, the meanings of the words and a text-to-speech module to pronounce the word. ARET also has an assessment tool that asks the user several kinds of questions to evaluate reading comprehension.

Our system currently targets English and offers a wider set of functionalities to users, in addition to a software architecture which can be extended very easily with more annotation components complying with UIMA interfaces. However, *our system architecture is language-independent*; adopting new languages is a fairly easy process as long as the relevant annotation tools and their UIMA interfaces are available.

3 User Functionality

From a reader's perspective, our tool is a web-based browser application. It runs in a multitude of browsers ranging over various platforms including touch tablets. It has a intuitive web interface to sign up, sign in, and browse available texts in the system's library. The reader has the option either to select a text from the library to read or to upload text she wants to read using the tool by including it in the library. If the reader chooses to submit her own text, the submitted text goes through several stages of real-time annotations that are used by the tool to make the text interactive. The tool then opens the text in a distraction-free tab.

The reader can interact with the text either by clicking on a word or selecting any segment of text. The system in turn takes into account the clicked/selected word's/segment's contents and its annotations by querying the server, highlights the segment (or something slightly and meaningfully larger, depending on the context) and presents a response, which most likely fits the reader's intent at the click position, as a default answer, along with a menu of other options. For instance,

- if the reader clicks on a content word, its meaning will be the most likely information she wants to know about i.e., the system presents the word meaning as the default response.
- if the reader clicks one of the words making up a named-entity, the system will extend and highlight the whole named-entity and present its type (e.g., person, location, etc.)
- if reader clicks on a pronoun, the system will display to who/what this pronoun refers by highlighting both the pronoun and the antecedent in context.
- the reader can explore beyond the default response by using the additional menu items provided: for instance she may ask about the grammatical role of a word in the sentence or get a list of questions involving a named entity and then select one and get it answered.

The tool provides all the available information to the reader but it orders these options according to an intention recognition module based on the annotations at the selected position. In the following sections, we describe the relevant details of the basic functions that our system provides.

3.1 Lexical Information

The current application provides the reader with the ability to inquire about lexical information such as word meaning, word type, sentence examples including the inquired word. Clicking on a word is the easiest and fastest way to access all the lexical information that is available for this word. In order to provide this lexical information we are making use of several tools which are fairly mature and can be used off-the-shelf.

Content Words: While there are many studies in second language acquisition on providing vocabulary and reading assistance (Prichard, 2008) and (Luppescu and Day, 1992). These studies showed that dictionaries can help in improving comprehension and efficient vocabulary acquisition. Luppescu and Day showed that the readers who use a printed dictionary have improved comprehension and acquisition, but negatively affect their reading speed.

Our tool provides vocabulary assistance to learners of English as a Second Language (ESL). When the reader selects a content word from the text, the tool provides the reader with the word definition and sentence examples including this word. We use WordNet (Fellbaum, 1998) as a broad-coverage machine-readable dictionary of English. Many words in WordNet have more than

one sense. Currently, we incorporate morphological analysis, part-of-speech filtering to narrow down the available senses and then present the user with the first WordNet sense under the selected part-of-speech, as shown in Figure 1.

Phrasal Verbs and Compound Nouns: Multiple word expressions may include phrasal verbs (e.g., reach into) and compound nouns. The meaning of these types of expressions often differ considerably from that of the underlying verb/noun and maybe unfamiliar to non-native english readers, and so they may interfere with content comprehension. In case the reader inquires about a word which is a part of a (possibly discontinuous) compound verb/noun, the tool highlights the whole compound structure and provides its meaning and also the meaning of the clicked word in case that the reader is interested in this specific word. Figure 2 shows the response to the reader on clicking the word *break* which is part of compound verb *break through*.

Function Words: Function words such as *though*, *whether*, *beyond*, etc., and other functional elements such as prepositions and determiners, can be confusing to a non-native English readers (Felice, 2008) . For function words (other than pronouns), the tool provides the reader with the word type, the part-of-speech of the word with some additional explanation. Figure 3 shows an example when the reader selects a function word.

Named Entities: One important function our tool provides, is identifying named-entities in the text. If the reader clicks/selects a name or part of it, the full span of the named entity is highlighted along with its category as shown in Figure 4.

Pronouns and Coreference Resolution: If a reader clicks on a pronoun, our tool presents the reader with the nearest previous named-entity for the pronoun and provides menus to navigate all previous and future coreferences. This would help the reader use nonlinear reading strategies and facilitate the extraction of information about the selected named entity through the document without reading through the whole text. Thus the reader can get an immediate flashback to the first time the person was encountered so she can re-read or remember more about this person, or see nearby references to get more recent context, and when done can snap back to the query point and continue reading. See Figure 5 for a sample interaction possibilities with pronouns.

3.2 Syntactic Information

Sometimes understanding the words meaning are not enough to fully understand the sentence. In order to help the user to understand the grammatical relations in a sentence, our tool provides the reader with the ability to inquire about the grammatical role of a word within the sentence. The sentences in the documents are previously annotated with dependency relations and when a word is clicked, one of the other menu items the user is presented with is the option to view the grammatical role of the word (shown with the button "Role" in the figures). When requested, we present the grammatical role in a user-friendly fashion by mapping dependency labels to more descriptive and meaningful labels as shown in Figure 6.

3.3 In-text Question Answering

Sometimes the reader may want to learn additional information about a named entity. Asking questions and getting answers may help in comprehension of the text and is a good way to get a flashback about the selected entity. If the user clicks on a named entity or a pronoun referring to it, the tool provides the reader with a short list of related questions that are automatically generated (at annotation time) involving the selected/referenced named entity, from previous sentences in the text. These questions are then ranked based on length, proximity and whether or not it or its answer involves another named entity, and a short list of questions are presented to the user. The user can then click on a question she is interested in, and immediately get the corresponding answer, *which is also generated at annotation time in parallel with question generation*. Figure 7 shows an example of this functionality.

3.4 Other Functionalities

Text summarization has been used to improve reading comprehension (Dermody and Speaker Jr, 1999) as well as document understanding (Wang et al., 2008) since it reduces information overload and provides a reader with a concise and informative text. Our tool provides the reader with a different levels of text summarization such as paragraph, multi-section, chapter and whole document summarization. The reader can select one or more paragraphs and ask the tool to summarize it for her. She can also ask the tool to summarize all the text before her selection which helps her to refresh her mind with the highlights of the preceding text. For this purpose, we use the Mead toolkit (Radev

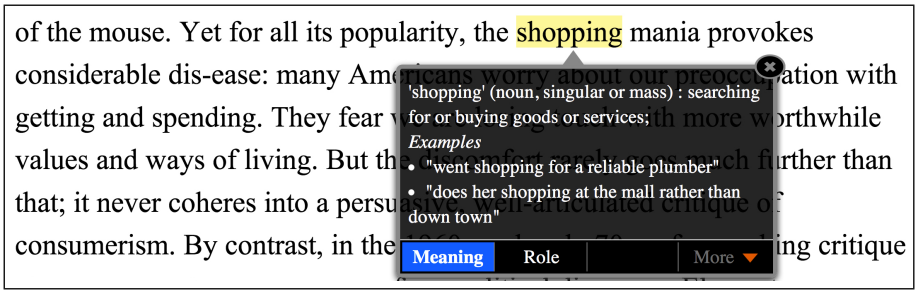

Figure 1: Looking up content word meaning

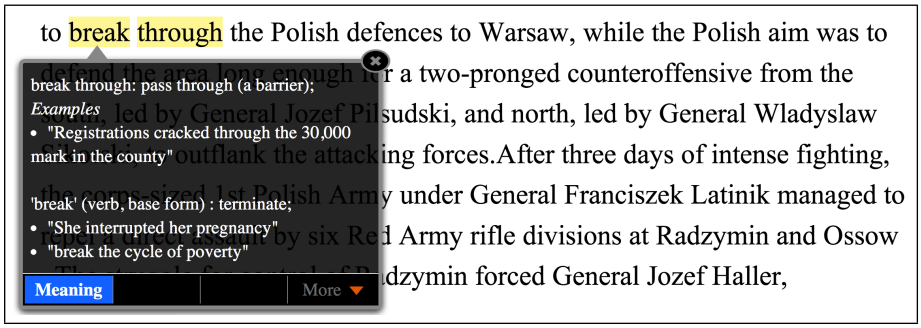

Figure 2: Response to selecting a compound verb

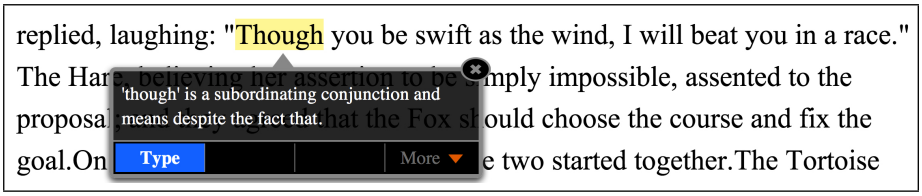

Figure 3: Response for a function word

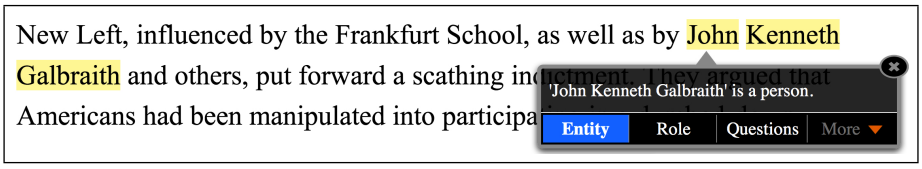

Figure 4: Response to selecting a portion of named-entity

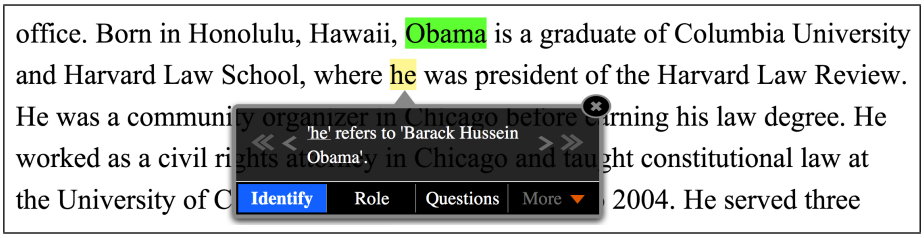

Figure 5: Identifying and Tracking named-entity mentions

et al., 2004) for English to provide the summarization functionality.

Another useful feature the tool also provides is logging the queries performed by a user together with data presented in response to the queries. At anytime, the reader can review the words she had problems with and asked about.

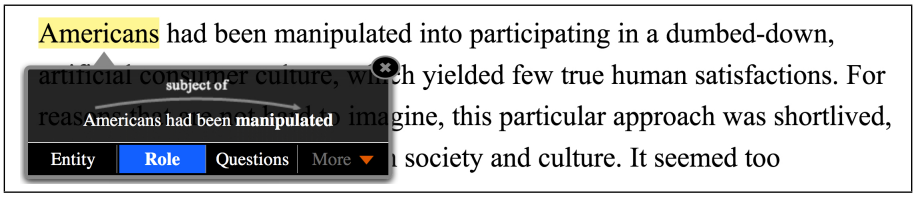

Figure 6: Showing the grammatical role of a word within the sentence

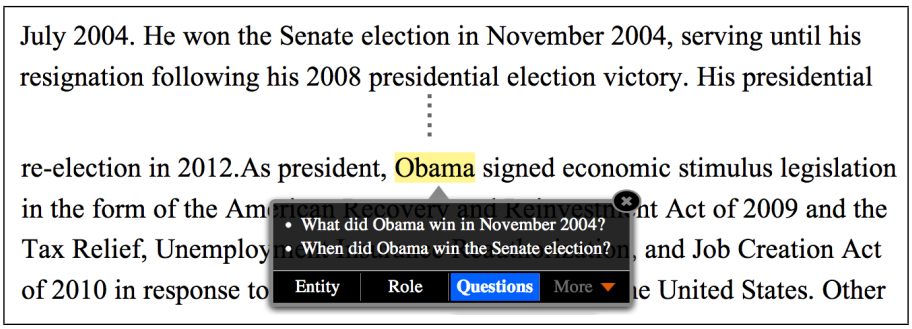

Figure 7: Presenting questions involving a named entity

4 System and Software Architecture

Our system follows a client-server paradigm where the server is responsible for all NLP-functionality, enriching plain text with annotations and retrieving them , while the client receives a version of the text that the user can interact and query the server with. The client here is a standard web browser, that can be accessed from browsers running on PCs or tablets, so on the reader's side no additional software is needed. The server processes and responds to requests received from these thin-clients.

All annotations that are needed to respond to user requests (except for summarization), are stored within a UIMA file produced by our annotators. The UIMA framework facilitates developing and integrating different text analysis engines and annotators in an extensible way and provides very powerful querying and search mechanisms for retrieving the annotations of the annotated documents.

4.1 Client Side

On the client side, the presentation layer is responsible for (i) keeping track of the user status and the opened documents, (ii) displaying the opened documents (iii) handling user-interactions, and (iv) sending queries to the server. The presentation layer is designed to be light and fast, with all the heavy processing to be done on the server side.

4.2 Server Side Query Processing

On the server side, the server receives requests and passes each request to the corresponding handler. These handlers in turn make use of two main units: the **data manager**, is responsible for all the database interactions on different data, the **query processing unit**, is responsible for extracting and reordering all the information related to a user query.

All documents in the system's library are all annotated with a series of NLP annotation tools and stored as a UIMA file. When UIMA is queried with a character position, it returns efficiently all the annotations associated with the word overlapping with that position which are then interpreted by the query processing unit.

During annotation, we segment the text into sentences, tokenize and run a POS tagger using Stanford CoreNLP.[1] We then use the following NLP components with appropriate UIMA wrappers to annotate our texts:

Stanford Dependency Parser (De Marneffe et al., 2006), provides grammatical relation annotations.

Stanford Named Entity Recognizer and **Stanford Co-reference Resolution** (Lee et al., 2013; Lee et al., 2011; Raghunathan et al., 2010) are used to determine the entities in the text and the relationships between them.

[1] http://nlp.stanford.edu/software/corenlp.shtml

Word Sense Annotator currently assigns the most frequent WordNet senses to content words by filtering the senses by just using the POS tag.

Compound Annotator identifies the phrasal verbs and the compound nouns in the text and adds additional annotation to words of a compound.

In-text Question Answering Annotator assigns the questions to the related named entities, and ranks them. The questions are generated using Heilman's question generator tool (Heilman and Smith, 2010).

For more details on the use of UIMA and the server architecture, please see Azab et al. (2013).

5 Evaluation

As we are using many tools and resources that have been developed for use on usually one genre of text, it will be an interesting experiment to see how they perform on the texts we will select for our library. We are currently in the process of preparing several short test documents for intrinsic evaluation of the performance of the annotation tools and reporting on their recall and precision. Manual evaluation of some of the components for one such document of about 1000 words is presented in Table 1.

We are also planning an extrinsic evaluation of the tool by having a group of non-native English speaking students use it and evaluate their experience. We are working together with a colleague who delivers a critical reading course who has provided us with a set of texts that students can read using our tools. He will then construct several evaluation experiments to see if our tool helps the students or not.

	Precision	Recall	F-score
NER	0.909	0.869	0.888
POS Tagger	0.986		
Coreference Resolution	0.679	0.63	0.653
Word Meaning	0.861	0.831	0.845
In-text Question Answering	0.62		

Table 1: Intrinsic evaluation of different NLP tools used.

6 Ongoing Work

We are currently working on improving our word sense identification annotator and implementing an additional sentence level annotation components:

Word-sense disambiguation : Word-sense disambiguation is a notoriously difficult problem and systems developed over the years have not been able to significantly exceed the most-frequent sense heuristic. Our current plan is to incorporate multiple word-sense disambiguators (e.g., Pedersen and Kolhatkar (2009)) along with super-sense taggers Ciaramita and Altun (2006), to build a system combination that can hopefully do a better job than the baseline, at least on our intrinsic test sets.

Lexical simplification : Text simplification can be defined as any process that reduces the syntactic or lexical complexity of a text while attempting to preserve its meaning and information content. The aim of text simplification is to make text easier to comprehend for a human user, or process by a program (Siddharthan, 2004). Text simplification has been studied for both human text readers and programs that process text. We are specifically concerned with students who try to acquire English as a second language (Petersen, 2007). Approaches for this target audience use simplification techniques as a preprocessing step to reduce complexity of sentence, mainly with respect to syntax (e.g., sentence decomposition on subordinate clause) and discourse structure (e.g., coreference resolution).

We are developing a sentence simplification module that addresses both lexical and limited syntactic simplification problems to help improve reading skills of non-native English learners. Our current focus is on developing a lexical simplification module that can identify the "difficult" vocabulary items or idiomatic uses in text, and annotate with their simpler versions.

7 Conclusion

We have presented our tool for helping non-native readers of English text to overcome language related hindrances while reading text. Our tool is also a showcase of English NLP and resources that have been built by the NLP community, integrated into an e-book reader application that can be adapted to more languages, provide resources are available. Our tool is based on a client-server software architecture, with the UIMA-framework being used for both annotation of documents and querying of annotations based on textual selections from the client applications running in browsers. We are also in the process of planning a test deployment for students for extrinsic experimentation.

Acknowledgments

This publication was made possible by grant NPRP-09-873-1-129 from the Qatar National Re-

search Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Mahmoud Azab, Ahmed Salama, Kemal Oflazer, Hideki Shima, Jun Araki, and Teruko Mitamura. 2013. An English reading tool as a NLP showcase. In *Proceedings of IJCNLP – System Demonstration*, Nagoya, Japan.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of EMNLP'06*, EMNLP '06, pages 594–602, Stroudsburg, PA, USA.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Margaret M Dermody and Richard B Speaker Jr. 1999. Reciprocal Strategy Training in Prediction, Clarification, Question Generating and Summarization to Improve Reading Comprehension. *Reading Improvement*, 36(1):16–23.

Soojeong Eom, Markus Dickinson, and Rebecca Sachs. 2012. Sense-specific lexical information for reading assistance. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 316–325, Stroudsburg, PA, USA.

Rachele De Felice. 2008. *Automatic Error Detection in Non-native English*. Ph.D. thesis, St Catherines College, University of Oxford.

Christiane Fellbaum. 1998. WordNet: An electronic lexical database. *The MIT Press*.

David Ferrucci and Adam Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

Marie-Jose Hamel and Marie-Christine Girard. 2000. FreeText - an advanced hypermedia CALL system featuring NLP tools for a smart treatment of authentic documents and free production exercises. In *Proceedings of EuroCALL-2000.*

Michael Heilman and Noah Smith. 2010. Extracting simplified statements for factual question generation. In *Proceedings of the 3rd Workshop on Question Generation*.

Michael Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. 2006. Classroom success of an intelligent tutoring system for lexical practice and reading comprehension. In *Proceedings of the Ninth International Conference on Spoken Language Processing*.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll shared task. In *Proceedings CONLL'11*, pages 28–34.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, pages 1–54.

Stuart Luppescu and Richard R. Day. 1992. Reading, Dictionaries, and Vocabulary Learning. *Language Learning*, 43(2):263–279.

Mohamed Maamouri, Wajdi Zaghouani, Violetta Cavalli-Sforza, Dave Graff, and Mike Ciul. 2012. Developing ARET: an NLP-based educational tool set for Arabic reading enhancement. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 127–135, Stroudsburg, PA, USA.

Jack Mostow and Gregory Aist. 2001. Evaluating tutors that listen: An overview of project listen. In K. Forbus andP. Feltovich, editor, *Smart Machines in Education: The coming revolution in educational technology.*, pages 169 – 234. MIT/AAAI Press.

John Nerbonne, Lauri Karttunen, Elena Paskaleva, Gabor Proszeky, and Tiit Roosmaa. 1997. Reading more into foreign languages. In *Proceedings of ANLP'97*, pages 135–138.

Ted Pedersen and Varada Kolhatkar. 2009. Wordnet::senserelate::allwords: a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of HLT/NAACL'06, Companion Volume: Demonstration Session*, NAACL-Demonstrations '09, pages 17–20, Stroudsburg, PA, USA.

S. Petersen. 2007. *Natural language processing tools for reading level assessment and text simplification for bilingual education*. Ph.D. thesis, University of Washington, Washington, USA.

Caleb Prichard. 2008. Evaluating l2 readers vocabulary strategies and dictionary use. *Reading in a Foreign Language*, 20(2):216–231.

D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang. 2004. Mead - a platform for multidocument multilingual text summarization. In *Proceedings of LREC 2004*, Lisbon, Portugal.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of EMNLP'10*, pages 492–501.

Advaith Siddharthan. 2004. Syntactic simplification and text cohesion. Technical Report 597, University of Cambridge.

Dingding Wang, Shenghuo Zhu, Tao Li, Yun Chi, and Yihong Gong. 2008. Integrating clustering and multi-document summarization to improve document understanding. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 1435–1436, New York, NY, USA. ACM.