CACLL
Computer Aided Colloquial Language Learning

Pam Leckie (leckiepa@seas.upenn.edu)
Feyisola Ogunfemi (feyisola@seas.upenn.edu)
Advisor: Ani Nenkova
Abstract

In this senior design project, we solve part of the problem of teaching common speech to second language English learners by focusing on idioms. We chose to focus on idioms because their figurative nature make them difficult to understand to second language learners, as their meanings can not be deciphered though a literal translation. Thus in order to understand idioms, the user must have prior knowledge of their meanings and their appropriate usage. It is difficult to learn more idioms through existing tools, such as an idiom dictionary. Dictionaries are only useful if the user knows what idiom they are looking for, as they are only searchable by idiom. However, this does nothing to help the user learn new idioms. We plan to incorporate this idea of an idiom dictionary, but alter it to allow users to expand their knowledge of idioms and appropriate usage.

There are two main aspects to our project. The first is a browser of idioms and definitions, much like existing dictionaries. However, the system also returns example paragraphs found in the New York Times. By reading the idioms in context, the user will learn more about appropriate idiom usage. This is much better than existing dictionaries that only show one example sentence, where one sentence may not be sufficient for complete comprehension. The second aspect is the main differentiating factor in our approach - the focus on keywords in definitions. Through various algorithms, we derived a list of keywords for each idiom based on the definition. When a second language learner has an idea on what he would like to say, this sort of approach is very helpful in finding creative ways of saying it. Using keywords to organize the idioms allows us to present the data in a way that is more suitable for learning. By viewing sets of related idioms at a time, the student would be able to understand the concepts and learn more at once. The system eliminates the need for a teacher and allows this to be a stand alone tool to teach this specific aspect of English. Our system is a teacher and has a better dictionary than current solutions, which makes it the perfect teaching aid for a second language English learner.
1. Related Work

Most English learning aides focus on proper grammar and syntax. However, a few attempt to teach colloquial style English.

1.1 Collocations

One system automatically assists second language learners with correct collocations (Chang et al., 2008), which helps improve colloquial English. The system takes in a written sample, checks a reference corpus for any matching collocation patterns, and gives suggestions of collocation usage. Although the system's process is similar to our design, it only focuses on verb-noun collocations. Additionally, the system was specifically designed to match between Chinese and English only. Our system differs in that it is designed to improve proper usage of idiomatic expressions and designed for second language learners of any first language background.

1.2 Error correction and examples

Another system uses the web for error correction (Gamon et al., 2008). This system is based on transcribed speech. The suggestion provider captures the potential error in the input sentence, while the example provider queries the web for example sentences. However, there has been no study on the usefulness or correctness of these example sentences. Our system queries articles from the New York Times, a valid source of data that is grammatically correct and of appropriate difficulty level, rather than random data from the internet in order to provide useful example sentences to the user.

1.3 Idiom dictionaries

Online idiom dictionaries provide idioms and their definition and example sentence, such as English Club Idioms Dictionary and The Free Idioms Dictionary. These are only searchable by idioms. If a user wants to learn new idioms, typing in a keyword will only return an idiom containing the keyword, which most likely has no relation to its definition. Our system makes idioms searchable by meaning, which is much more helpful in teaching the user exactly what he is intending to say.

Another online tool, English Daily, attempts to teach users idioms. This site embeds the idiom and
the definition in a 3-5 sentence example. This site has a similar approach as ours in terms of providing examples to teach proper usage and assist in comprehension. However, there are only a select number of idioms provided (44). Our system covers more than 700 idioms and teaches users a much wider range of idioms.

2. Technical Approach

2.1 System overview

Our system performs two main functions. The first function allows users to search for idioms based on their meanings. Extensive experimentation was done through two types of keyword algorithms to define the best keywords for idioms. The second function allows the user to browse through idioms alphabetically and see multiple examples from the NY Times. These two functions help the user to understand idioms and how they should be used, and shows how they are different from formal English.

The combination of keyword searching and browsing allows for better learning than current systems such as the The Free Online Dictionary, which has just an alphabetized browser. While exploring the site, we discovered that this was problematic, as sometimes one idiom may begin with more than one word, for example “its raining cats and dogs,” and “raining cats and dogs.” Users may only think to search one form of such an idiom, and thus have a hard time finding it. This is alleviated through the keyword search as it is based on the meaning which remains the same between variations of an idiom, while the browser still allows the user to come across idioms that she may not have thought to search for.

We used java for all of our programs, since classes and inheritance help a lot with our search programs. We built and ran everything on our home Linux machines, as this allowed us to freely use open source software such as Wget, which was used to compile data from the internet, and the Netbeans IDE, which was used to create the GUI.

2.2 Browsing

Through the browser, the user can browse through a list of idioms, organized alphabetically by the first word in the idiom. Doing this allows the user to browse through many idioms as an alternative to a
very focused search. This component provides the best context for each idiom in the dictionary as the user is able to see many examples of its use in paragraphs from the NY Times corpus while browsing.

This component also has a “suggest an idiom” section, which allows the user to suggest keywords which they believe best exemplify the idiom which is currently displayed. This allows the system to pick up on synonyms of current words, or some good keywords which may have been missed by both the keyword finding algorithms. The newly suggested keywords are stored in a separate file, and are ranked higher than the keywords found by the tf-idf banking or log likelihood ratios. This was done in order to rank future search results such that idioms that corresponded to the suggested keywords are displayed before those that corresponded just with the keywords found from our algorithms. This is very important, because it allows the system to “learn” and improve. Since it is easier to trust human interpretation rather than computer interpretation, suggestions allow the system to collect keywords that are potentially more useful for learners of English as a second language. Comparisons can also be made as to how suggestions differ from keywords from native English speakers.

2.3 GUI

We created our user interface using the Netbeans IDE GUI builder. We selected this tool because it is open source, has a lot of documentation on usage, and is very similar to the C# GUI builder, which Fey is somewhat familiar with. The GUI is a very important part of our system, because it is what pulls all of the pieces of our project together into a single, compact, easy to use program. Fig. 1 illustrates how the system works.
Figure 1: System overview

The left panel of the system is the keyword search section. Here, a user types in keywords and if those keywords are in any of the two keyword files, then the idioms, along with their definitions, are printed in the box directly below the search box. If the keyword was ever suggested by the user, the idiom for which it was suggested for is displayed before other related idioms.

On the browser half of the system, the user can select a letter in the alphabet to browse through idioms beginning with that letter. The idioms and their definitions are displayed one by one in one box as well. Both sections of the system have an examples box where the top ten examples from the NY Times corpus, providing contextual examples, of each idiom are displayed. There is also a text box here, wherein a user can suggest a keyword for the idiom currently being displayed, and have that keyword added to the keyword list.
2.4 Finding Keywords

Two algorithms were used to define keywords for each idiom: log likelihood and term frequency – inverse document frequency (tf-idf). Log likelihood (Lin and Hovy, 2000) is a method that finds topic signature terms based on a collection of data. For our purposes, the algorithm was used on both the definitions and the example paragraphs to see which produced the best keywords. The second algorithm, tf-idf (Salton and Buckley, 1988), was also ran on both the definitions and examples.

2.4.1 Log likelihood of definitions

Let T be the definition of the idiom. Let NT be the collection of all definitions. Let t be a term. Then we define two hypotheses:

H1: \( P(t \mid T) = P(t \mid NT) = p \), t is not a descriptive term

H2: \( P(t \mid T) = p_1 \) and \( P(t \mid NT) = p_2 \) and \( p_1 > p_2 \), t is a descriptive term

\[
x = \frac{\text{likelihood of the data given } H_1}{\text{likelihood of the data given } H_2} = \frac{b(c_t, N, p)}{b(c_T, N_T, p_1) \times b(c_{NT}, N_{NT}, p_2)}
\]

where \( b(k, N, p) \) is the binomial distribution, or probability of observing term t appearing k times in N trials. Here, \( c_t \), is the times t appears in the entire corpus, \( c_T \) is the times t occurs in T, \( c_{NT} \) is the times t occurs in NT, N is the number of words in the entire corpus, \( N_T \), is the number of words in T, and \( N_{NT} \) is the number of words in NT.

Terms for which \( -2\log(x) > 5 \) were considered “topic signature terms.” Those with the top 10 values were used as keywords.

2.4.2 Log likelihood of examples

Let T be the paragraph examples for the idiom. Let NT be the entire NY Times corpus. Let t be a term. Use the same hypothesis and likelihood equation as was used for definitions, except that terms for which \( -2\log(x) > 10 \) were considered “topic signature terms” and top 10 were used as keywords.

2.4.3 Tf-idf Ranking Scheme

The second algorithm, the tf-idf ranking scheme, ranks words found in a document, or in this case, an idiom, relative to both its frequency in the idiom and the inverse of its frequency in other idioms, to produce term ranks which define how related the term is to a certain idiom. That is.
Term frequency (tf):
= number of times the term appears in the idiom and its definition.

Inverse document frequency (idf):
= \log (\text{number of idioms containing the keyword}/\text{total number of idioms in the dictionary})

Tf-idf Rank:
= \text{term frequency (tf)} \times \text{inverse document frequency (idf)}

This algorithm was run over our final dictionary of over 700 idioms to produce a set of keywords for each idiom. We then wrote a program to extract the five highest ranked keywords from the keyword list, and associate them with the idiom. In our initial trial, we included the idiom, definition and example in the “document” used to determine the idf, but later revised this to only take the idiom and definition into account. This worked because the examples always contained the idiom, which caused the words in the idiom to be ranked too high and because most other words in the example were very situational and did not relate very much to the meaning of the idiom.

We altered the tf-idf ranking scheme to be used over the collection of articles from the NY Times so that we could compare the results with the log likelihood algorithm, and determine which was better. By manual inspection, as show in Table 1, of the keywords returned by each method, we determined that the log likelihood algorithm produced the best keywords. However, this algorithm could not produce keywords for every example (because the definitions were too short) so our final system uses log likelihood keywords when they exist, otherwise it uses the keywords produced by the tf-idf ranking scheme, the second best method.
Table 1. Percentage of good keywords by algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>tfidf - definitions</th>
<th>tfidf - examples</th>
<th>log lik - definitions</th>
<th>log lik - examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIXED</td>
<td>40</td>
<td>10</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>once in a blue moon</td>
<td>40</td>
<td>0</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>ignorance is bliss</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>young blood</td>
<td>20</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>like a fish out of water</td>
<td>20</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>a ballpark estimate</td>
<td>20</td>
<td>10</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>fresh as a daisy</td>
<td>20</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>agree to disagree</td>
<td>80</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>more than meets the eye</td>
<td>40</td>
<td>0</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>forty winks</td>
<td>80</td>
<td>10</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>beat around the bush</td>
<td>20</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>NON-FIXED</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>draw a blank</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>give the green light</td>
<td>20</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>jump for joy</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>hold the fort</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>give it a whirl</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>know the ropes</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>keep your word</td>
<td>10</td>
<td>10</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>hold your head high</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>bite your tongue</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>give it your all</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>34</td>
<td>2.5</td>
<td>48.68</td>
<td>3</td>
</tr>
</tbody>
</table>

2.5 Data Collection and Management and example extraction

Our final dictionary of idioms, definitions and examples was collected from three online idioms dictionaries. We used the Wget program to download the data from the sites and then used regular expressions to parse through the html tags and reveal the main text. We did the majority of our initial research about idioms on the Free Online Dictionary (idioms.freedictionary.com) and found the most exhaustive list of idioms there, but had many issues when trying to download information from the site, since the webmaster forbade the use of wget on the site and eventually locked us out. The two dictionaries used for the final dictionary were The English Club Idioms Dictionary, and the Idiom Connection. Since the largest of these dictionaries, the English Club Idioms Dictionary, did not provide examples of the idioms, we derived examples of those idioms in use from the NY Times corpus.

Finding relevant examples was also a critical aspect to our system. In order to find useful examples, a corpus of over 1 million New York Times articles (1987-2007) was scanned. All files had to be
converted from xml to txt files first. Do to the large size of our database, in order to extract examples, Apache Lucene, a text search engine written in Java, was used to search for each idiom in the corpus of articles. Once a list of articles containing each idiom was obtained, the paragraph in which the idiom appeared was extracted. As the rest of the article is irrelevant for an example or keyword searching, only the paragraph was used in future algorithms. Additionally, a previously proposed idea was the use of multiple choice questions, or cloze items, to test the user’s knowledge of an idiom. We did not include this feature because a larger focus on using examples and meanings increased the user’s ability to learn, while cloze items simply increase the ability of an instructor to evaluate how much the student learned. We no longer want the system to be a teaching aid for English classes, but rather a reference tool for learners of English as a second language. Using New York Times examples gives the user relevant and modern examples to learn from.

3. Conclusion

We were successful in creating a system which is conducive to learning new idioms. We successfully acquired, organized and manipulated data from current idiom dictionaries to provide a way for learners of English as a second language to search for idioms of based on their meanings, get the necessary context to understand new idioms and to suggest new keywords for idioms whose definitions/keywords were confusing. During this process, we were able to apply NLP techniques of log likelihood and tf-idf to aid in our keyword discovery. We discovered that the tf-idf ranking scheme did not produce as favorable results as the log likelihood ratio algorithm did. We also found that these algorithms produced far worse results when run over data which included context and examples, rather than data including just the idioms and their meanings. We can conclude that the context surrounding the idioms do little to help define it in terms of automatic parsing.

As for future work, since we already created an easy to use interface and frame, developers with more robust searching or keyword discovery algorithms can build upon it to create even better solutions. For example, improvement on the keyword selection process, which we did not have time to do, could be
achieved by incorporating a thesaurus to find the synonyms of each keyword, then adding them to the keyword list, to make it more robust and exhaustive. We were successful in achieving the majority of our goals for the Computer Aided Colloquial Language Learning System.

4. Bibliography


This paper appears in the CALL journal by the authors who are experts in the field of Computer Science, English, and Foreign Languages and Literature. Having these background allowed the authors to give expert opinions not only on the design of the system, but the pedagogical nature of the language tool. This paper was extremely helpful in designing our system. The idea of using two corpora (reference & learner corpus) and the algorithm for parsing the user's input, searching for intended use, and returning useful examples are the backbone for our project.


This paper was presented at conference sponsored by the Association of Computation Linguistics. The authors are expert in the field of Computer Science. The system parses the BNC to discover verb-noun formed idiomatic expressions. This paper is especially important to our project in helping us determine which phrases in a given sentence can be grammatically considered an idiom. Although it only measures verb-noun idioms, it is useful for identifying patterns and lexical fixedness.


This paper was presented at the Conference on Empirical Methods in Natural Language Processing. The importance of this paper lies in determining ambiguous idioms. This is very relevant to our project, as many of the examples found in the NY Times can be considered literal or figurative representations of the idioms in context. The techniques outlined in this paper can help to ignore the more literal examples.


This paper appeared in Applied Linguistics, an Oxford journal which publishes research which publishes research on theory, practice, and study of language in specific situations in which people use and learn languages. Although the paper is not current, it gave insight in the study of language. As computer scientists, this paper helped explain more of the linguistic approach to our project. The most important aspect of the paper was the description of common second language learners' mistakes. This gives insight that will help our project in anticipating second language learners' errors.


This paper was presented at the International Joint Conference on Natural Language Processing. The authors are experts in Computer Science, and the system lies under research for Microsoft. This paper is useful since it deals with a transcribed speech corpus, since much of our learner corpora consists of transcribed speech (which differs than written English in some aspects). However, there is no data on the
usefulness of the examples derived by the system.


Corpus of English idioms drawn from four idiom dictionaries, found in the Linguistic Data Consortium.


This paper was presented at a conference for Interspeech under sponsorship from the International Speech Communication Association. The authors are experts in Spoken Language Systems. This article outlines a procedure for detecting complete grammar correction of second language learners' of English's sentences. Although we are not concerned with error detection, this paper is useful for our project in helping to parse ill-formed sentences. Our project will parse sentences to detect possible usage of idiomatic expressions, but it will need to be flexible to account user errors, since they will be second language learners.


This paper was presented at a conference for Interspeech under sponsorship from the International Speech Communication Association. The authors are experts in Spoken Language Systems. The paper describes the authors' attempt to automatically create answer choices for fill-in the blank (cloze item) questions that are difficult enough to cause confusion among test takers. This paper was useful to our project since we plan on using cloze items in our language tool. The most important aspect of this paper was the description of the algorithm for generating the answer choices.


This paper was presented at a conference for the Association for Computational Linguistics: Human Language Technologies. The authors are experts in Spoken Language Systems. The paper describes the authors' system to correct verb form errors made by non-native English speakers. This paper was useful to our project since it proposes a future problem that can be relevant to our project: while parsing written data, the system finds it difficult to process transcribed speech due to colloquial expressions. The particularly important aspect of this article is detecting semantic and syntactic errors and identifying intended syntactic relationships.


This paper was presented at a conference for Computational Linguistics. This paper presents a possible solution for automatic text summarization. For our purposes, the relevant part was the section on topic signature terms. We used the topic signature term algorithm in order to define keywords to idioms.


This paper was presented at an International Conference on Computational Linguistics. Tetreault is a representative of Educational Testing Service, a company which make heavy usage of automatic detection of errors and generation of cloze items. The paper was useful for its methodology of evaluating
the proposed system. This methodology can help us determine the effectiveness of our own system.


This paper was useful for using the tf-idf approach for finding keywords for idioms.


This paper was presented at the Association for Computational Linguistics's workshop on Deep Lexical Acquisition. Dorow is an expert in Natural Language Processing. The paper offers a unique algorithm for extracting idioms from a sentence. This will be important to our project in helping determine which examples returned by our system use the idiom in a literal or figurative manner.


This paper was presented at the International Joint Conference on Natural Language Processing. The authors use the web as a source of training data for their system. The system also queries the web, using correlation between user input and web hits as a means of error detection. Although this process may not be accurate, the paper is useful for defining ways of incorporating usage of data from web within our project.

Online idiom dictionaries:

EnglishClub – http://www.englishclub.com/ref/Idioms/
Idiom Connection - http://idiomconnection.com/mostfrequent