Combining Lexical Resources for Text Analysis of Game Walkthroughs

Michael Billot, University of Colorado at Colorado Springs

***Abstract*—One approach to text analysis is motivated by a desire to understand the actions that are most frequent within a body of text. By analyzing words in the text, primarily verbs, connections can be drawn to the actions that are described by the words. The trouble is that single words can have many meanings and evoke many different situations. For that reason, word sense disambiguation software is a vital part of this project. Lexical resources are also needed because they contain two major types of information; the meanings behind words and the relationships between meanings. Another resource needed by this project is a part of speech tagger, which is used for extracting important parts of speech to work with.**

I. Introduction

MANY lexical tools have been developed to assist with computational linguistics and natural language processing. These tools often overlap with their capabilities, but they differ greatly in the ways in which they structure their lexical data. Also, they each have their own strengths and weaknesses. Certain tools often don't cover the same data as other tools. Thus using two tools in conjunction can help populate sparsely covered areas in either or both tools. Using them together may mitigate disadvantages and weaknesses that occur in a single tool. This projects approach to text analysis is based on the assumption that multiple tools used in conjunction are more powerful than any single tool by itself.

II. Motivation

The purpose of this project is to process input text and identify semantic trends, and then use those semantic trends to help generate parameterized actions. Actions are represented by WordNet senses of verbs. An example is the verb open, which occurs fifty-three times in the text. The goal is to output the actions that most frequently occur with the use of the verb open. This is done for every verb that occurs in the text. By doing so, we can produce a resource that dynamically associates words with actions for a specific domain. A possible advantage of this resource is to help disambiguate commands into the actions, such as giving a command to a 3D agent.

III. Related Work

Mihalcea et al. researched various implementations of the PageRank Algorithm for word sense disambiguation, including a combined method that also used the Lesk Algorithm [1].

Banerjee performed word sense disambiguation using WordNet and an adapted Lesk Algorithm. The implementation of his approach comes across as being more simplistic than disambiguation approaches presented in other papers. In the end his disambiguation performance was relatively low in comparison to others [2].

Agirre and Soroa delve into a personalized PageRank algorithm (PPR) that mitigates certain words from having too high of weights [3]].

Giuglea and Moschitti explore ways to connect FrameNet and PropBank via VerbNet [4]. Their process revolves mostly around the parameterized structure of the lexical tools. They define relationships based on common parameterization.

Paziena et al. looked at ways to study verb relations by using WordNet, VerbNet, and PropBank [5]. Their approach was based on the combination of different relational knowledge that each tool provides. WordNet provides verb sense relationships. VerbNet, on the other hand, consists of verb-sense frame knowledge. Their goal was to produce examples of verb pairs that have semantic relations and specific predicate-argument structures.

IV. Approach

This projects approach to performing verb analysis of text is to combine different software and lexical resources. For lexical resources, using their lexical data in combination will strengthen the capabilities of each tool. The other software will allow us to work with the lexical resources in new ways.

A. Domain Used

The selected domain for this project is video game walkthroughs for The Legend of Zelda: Ocarina of Time. The reasoning behind this is that the walkthroughs are freely available online. Also the walkthroughs are full of commands that tell the reader what to do. The action verbs within these commands help the verb sense disambiguation process, whereas sentences with many helping verbs make it more difficult and complicated. Helping verbs are more difficult to disambiguate because their usage in the sentence often does not align well with senses in WordNet.

B. Tools Used

The tools used for this project include the lexical resources WordNet and FrameNet. Another tool is a word sense disambiguator named UKB. Lastly, the Stanford NLP Part of Speech Tagger is used.

1) FrameNet: FrameNet categorizes the English lexicon into semantic frames, which describe situations and the words which compose them [6]. This relationship between a word and a meaning is called a lexical unit. Frames are often associated with multiple lexical units. Within the frame there are frame elements, which are references to supporting frames. Frame elements are divided into two categories: core and non-core frame elements. Core frame elements are mandatory parameters for a frame, and non-core frame elements are optional constraints to the frame. Ultimately, semantic frames correspond to situations, lexical units correspond to the words that evoke those situations (often verbs), and frame elements correspond to the syntactic dependents in a sentence.

2) WordNet: WordNet organizes words into structures called synsets. Synsets encapsulate synonomous words and inter-synset relationships [7]. These synset relationships point to lexical relations of word form, as well as semantic relations of word meaning. Types of relations include hypernymy, hyponymy, antonymy, holonymy, and meronymy.

3) UKB Word Sense Disambiguator: UKB performs word sense disambiguation by using an implementation of the PageRank algorithm [8]. The PageRank algorithm works by ranking nodes, which are normally websites. In this case the nodes are WordNet senses. The reputation of each node is affected by the reputations of nodes that point to it. Therefore, a node with influential nodes pointing to it will have a stronger influence on the nodes that it points to. The following image does a good job illustrating this process.

4) Stanford NLP POS Tagger: The Stanford NLP POS Tagger uses one of two models for English part-of-speech tagging. The first is a model trained on sections 0 through 18 of the Wall Street Journal, uses a left3words architecture, and is 88.85% correct on unknown words. The second model is trained on the same WSJ sections, uses a bidirectional architecture, and can correctly tag 90.46% of unknown words [9]. Either model will be adequate for this project, yet the model using the left3words architecture may be a better option because it takes significantly less time to part of speech tag sentences.

C. Steps

The two primary goals are to sense disambiguate verbs and to align senses with frames. There are a few steps involved to accomplish these two goals.

1) Part of speech tag each sentence in the text.
2) Extract parts of speech from the tagged sentences and store verb frequencies and verb senses in memory
3) Generate context groups for each sentence that contain the verbs, nouns, adjectives, and adverbs of a sentence.
4) Feed the context groups into a word sense disambiguator to get the best verb senses.
5) For the verbs that occur in the text, align their senses with frames in FrameNet.
6) Write the parsed information to a file. This info includes verb frequencies, verb sense IDs, sentences, context groups, and extracted parts of speech.
7) Load the parse information, verb disambiguation results, and frame-sense alignment results into a GUI program to review the results.

1) Word Extraction: After the sentence has been part of speech tagged then the key words must then be extracted from it. This collection of words essentially represents the sentence. There are two types of undesirable collections of words. The first is a sentence that has no verbs identified, either because the part of speech tagger failed or it was never a complete sentence to begin with. This type of sentence is useless because it has no verb to disambiguate. Another undesirable case is where there are less than three parts of speech in the sentence. If there isnt enough contextual information, then verb sense disambiguation cannot perform accurately. Otherwise, if there is enough contextual information, then a context group is made with the sentences parts of speech. The context groups are then passed to the UKB word sense disambiguator.

2) Verb Sense Disambiguation: Word sense disambiguation is performed solely on the verbs of the text. However, the nouns, adjectives, and adverbs will help with this process. UKB will perform verb sense disambiguation using graph-based and lexical similarity methods using a pre-existing WordNet knowledge base [8]. These methods are founded on the PageRank algorithm, variations of which are explored by Agirre and Lopez [10]. Verb sense disambiguation in particular is more difficult and less accurate in comparison to nouns and adjectives [3].

A primary goal of this project is to figure out methods to increase the verb disambiguation performance. An example would be to figure out what is keeping UKB from getting the right answer. After finding what it needs to generate the correct answer, the missing parts can be added to the context group. This process is sort of like nudging UKB in the right direction. Also, one could ask the reverse; what is leading it to the wrong answer? There are sometimes words or word relations that cause UKB to favor one sense. If these could be identified, then they could be manipulated or removed. Lets face it, WordNet has a very high level of granularity. This causes it to have a surplus of word-sense relationships, some of which contribute to improper sense disambiguation.

3) Frame-Sense Alignment: Frame-sense alignment involves aligning verb senses with the FrameNet frames that describe them. This process is not dependent on the results of verb sense disambiguation, since frame alignment works with all senses individually.

A method inspired by Burchardts frame assignment system will be used. As Burchardt suggests, frame assignment may not just involve a single word, but synonyms and hypernyms of that word [11]. The first step involves generating a set of WordNet relatives of the target sense, which includes synonyms, hypernyms, and antonyms. Each relative word then helps score frames. If a relative word evokes a frame, then that frame has the following added to its score: $(framesEvoked(relativeWord))^{-1}$.

If a WordNet relative word evokes many frames, then it has a higher probability of ambiguity. This is taken into account by dividing by the number of frames evoked. Doing so renders ambiguous words to have less of an impact on the scoring of frames. Therefore, relative words that evoke fewer frames are more valuable in the scoring process.

To select the best frame, each frames scores are summed up and the frame with the highest score is the winner. Other methods that measure lemma relatedness to a frame are experimented by Nuges and Johansson [12].

For certain verb senses there are not enough relative words to deem the best frame. If there is no winning frame, then the sense will be considered to be frameless. Another unique case is where two or more frames tie. In this case all the frames will be considered to be the best frames. Senses that perform poorly in frame-sense alignment are in general less common in English.

D. Output

The final output will show semantic information about the input text. Most of the information will regard the verbs within the text. Here is a list of the primary displayable data.

- Frequencies of verbs
- Frequencies of disambiguated verb-senses
- Frequencies of actual verb-senses
- Accuracy of verb sense disambiguation
- Best fit frames for each sense
- Pronunciation of words using dictionary.com

The processes of disambiguating verbs and aligning senses to frames is very time consuming, especially for verbs that have a high number of relationships in WordNet. For that reason the project consists of two programs. The first is a program that generates all the information, and the second is a GUI program that displays it.

E. Metrics

There will be two main metrics used to assess the success of the project. The first of which will examine the accuracy of the verb-sense disambiguation. The second will examine the accuracy of the frame-sense alignment.

1) Sense Disambiguation Metrics: In order to determine the disambiguation accuracy one must manually tag each instance of a verb with its real sense, if any. A disadvantage to accuracy occurs because usually a verb is sense tagged by a single person. A remedy would be to have multiple people disambiguate the same sentence in order to assure a more accurate metric. However, it is costly enough to sense tag each verb by hand, so at this point each verb is sense tagged by one person.

After all of the instances of a verb have been disambiguated both by UKB and by hand, then we have the sense disambiguation accuracy. The accuracy information includes the percentage helping verbs with no fitting sense, and the percentage of correct and incorrect disambiguated senses. The following image is an example showing the results for the verb see.

```
see occurs 57 times. Polysemy count: 24
0/57: 0.0%: Helping verbs with no fitting sense
12/57: 21.052631%: Incorrect sense assignments
45/57: 78.94737%: Correct sense assignments
```

2) Frame Alignment Metrics: Frame alignment accuracy is determined manually. If the best fit frame accurately matches the action going on in the sentence, then it is deemed to be correct. A better approach would be to use an existing mapping between WordNet and FrameNet. Unfortunately, there are no available mappings between their current versions, WordNet 3.0 and FrameNet 1.2.

The frame alignment metric provides an accuracy measurement for each verb. It is designed to treat the more prominent senses of a verb as being more important. The first sense of a verb is always the most frequent according to sense annotated data. Therefore, its correct frame alignment should be more important than correctly aligning an uncommon sense.

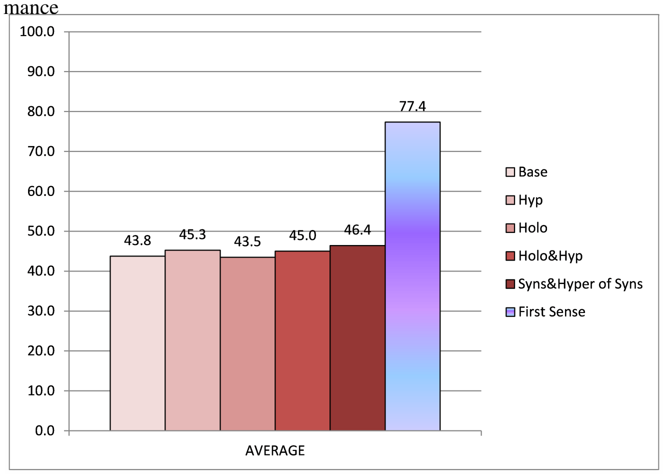
Say a verb has a polysemy count of n. Its highest possible score is $0.5\left(n^2+n\right)$, which is equivalent to $\sum_{i=1}^{n} i$. If the first and most common sense is aligned to the correct frame, then it receives n points. If the second is correct, then it receives n-1 points. If the least common sense is correct, it receives 1 point. The points a verb scores divided by its highest possible score gives its sense-frame alignment accuracy.

V. Results

The results to the projects experiments fall into two separate areas. The first is sense disambiguation results and the other is frame-sense alignment results.

A. Sense Disambiguation Results

Fig. 1. Average Personalized PageRank Verb-Sense Disambiguation Performance



Sense disambiguation results are based off of an analysis of the top ten most frequently occurring verbs. These verbs are use, kill, hit, see, open, shoot, walk, find, and fall. There are some verbs that are frequent but excluded from the results. One such verb is be, which is used as a helping verb extensively. Other common verbs that are frequently used as helping verbs frequently include get, go, and do. When verbs are used as helping verbs they are usually difficult to disambiguate because they dont always have a fitting WordNet sense.

There are disambiguation results for six different methods. Five of them use the Personalized PageRank algorithm supplied by the UKB software. They differ in the way that they construct context groups. The base method simply disambiguates the verbs based on the verbs, nouns, adjectives, and adverbs that are in the same sentence. The other methods include WordNet relatives of the nouns in the context group. These relatives are hypernyms, holonyms, and synonyms. A final disambiguation method is called the first sense method. It always assumes a verb is being used in its first sense.

There is a reason for using WordNet relatives of nouns. Nouns are sometimes too vague or unrelated to the verb. The hypernyms and holonyms of nouns could make the context groups more descriptive. For example, in the text there are many instances of the verb kill. The text usually talks about killing monsters in game. Adding in hypernyms of what is being killed may help the Personalized PageRank algorithm develop a stronger connection to a specific sense of kill.

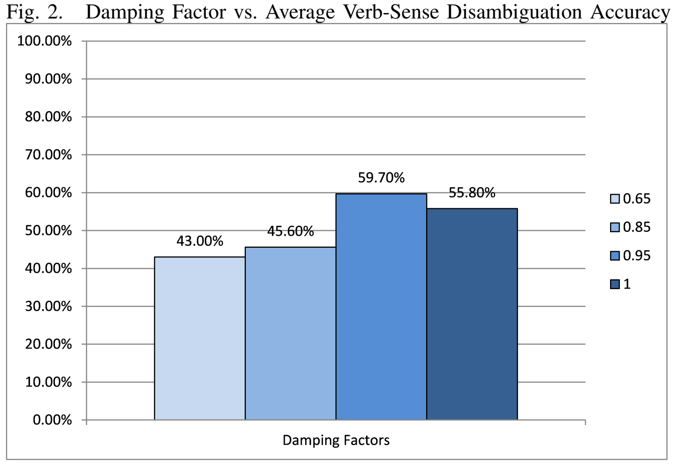
There are also disambiguation results from testing different damping factors. The default damping value is 0.85, which is the recommended value. However, changing the damping value does produce some effects. The lower the damping factor then the faster the iterations will converge on a sense[13]. With a high damping factor, nodes will increase their page rank more quickly. It will be interesting to see how the damping factor influences sense disambiguation.

1) 5.1.1 Average VSD Performance of Different Methods: The following graph shows the average sense disambiguation accuracy of the six different methods. The methods are the following:
1) Base: Context group consists only of verbs, nouns, adjectives, and adverbs.
2) Hyp: Hypernyms of nouns are added to the context group.
3) Holo: Holonyms of nouns are added.
4) Hyp&Holo: Holonyms and hypernyms of nouns are added.
5) Syns&Hyper of Syns: Synonyms and hypernyms of synonyms are added.
6) First Sense: The verbs first sense is always chosen.

Figure 1 shows that verb disambiguation performance does not change drastically when adding WordNet relatives to the context groups. In comparisson to the base results, the addition of WordNet relatives created a 2.6% performance increase at the most. The best performing methods, albeit a small performance advantage, were the ones that included hypernyms. On average, adding hypernyms caused a 1.8% performance increase. Future tests will be done to test other combinations of WordNet relatives. One particular test is adding in WordNet relatives of the verb itself. Potentially, a certain combination of WordNet relatives could cause a performance increase a few percent higher.

One method with a staggering success rate is the first sense method. This method performs so well because the first sense is usually the most common by far. On average it performed 72.6% better than all the other methods. For this reason, it is likely that a good apporach to verb sense disambiguation may involve weighing the first sense of a verb more heavily than the other senses.

Fig. 2. Damping Factor vs. Average Verb-Sense Disambiguation Accuracy



2) Damping Factor : Three different damping factors were tested in addition to the default damping factor of 0.85. The other tested values were 0.65, 0.95, and 1.0. The graph below shows the average sense disambiguation performance of the ten frequently occurring verbs.

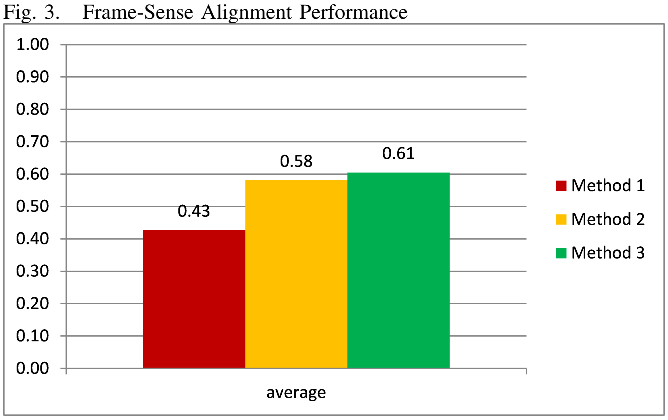
It appears that the default damping factor for the personalized pagerank algorithm may not be the best option for verb disambiguation accuracy. Not even the documentation for the software recommends using a different damping factor. The documentation only mentions that the default damping factor is 0.85. However, based of of the 14.1% increase between the 0.85 and 0.95, one can conclude that there is an optimum damping factor.

The most accurate damping factor is not necessarily the highest possible damping factor. The damping factor of 1.0 actually performed less accurately than the factor of 0.95. In the future it would be nice to test a range of damping factors with small increments of 0.01 or 0.02. That way an optimum damping factor for a body of text could be identified. Other text documents could be analyzed and their optimum damping factors could be compared. Judging by the results expressed in figure 2, the optimum damping factor is probably somewhere around 0.95.

B. Frame Alignment Results

There are three different methods for performing frame-sense alignment. All the methods are based off of using different WordNet relatives of the senses. The three different methods use the following WordNet relatives to score the best frame. Note: frame-sense alignment will not be continued in

Fig. 3. Frame-Sense Alignment Performance



future research. See the future research section for details.

Method 1: Synonyms, hypernyms
Method 2: Synonyms, hypernyms, hyponyms, antonyms
Method 3: Synonyms, hypernyms, hyponyms, antonyms, holonyms, meronyms

The difference in accuracy between method two and method one clearly shows an increase in frame-sense alignment performance because of the additon of hyponyms and antonyms. When comparing methods two and three, the results do not demonstrate a significant impact because of the addition of holonyms and merynyms. Upon further investigation, it looks like verbs usually have no holonymy or meronymy relationships.

VI. Future Work

In future work, there will be some drastic changes to this project. First of all, sense-tagged text will be used. In the event that there isn't enough sense tagged text to produce definitive results, then we may have to resort back to hand tagging verb senses. There are a couple of sense-tagged resources that could be good candidates for future resources. The first is sense-tagged text from the Senseval3 competition [13]. One potential problem is that this data is tagged using WordNet 1.7 senses, and the most current version of WordNet is 3.0. A mapping between WordNet 1.7 and 3.0 would have to be used in order to effectively use this resource. Senseval is currently in its firth competition. The fourth and fifth competitions might also provide sense-tagged text. In end, if there is not enough sense-tagged data to work with, then hand tagging will have to be done.

Another change to the project is the exclusion of FrameNet and frame-sense alignment. Frame-sense alignment was originally included to assist with the long term goal of parameterized action representation. However, that goal is unachievable right now because of how inaccurate verb sense disambigation is. For that reason, the sole approach in the future will be to improve verb sense disambiguation performance.

Future work will involve verb sense disambiguation using the Personalized PageRank method provided by UKB. The goal is to optimize Personalized PageRank's as much as possible. UKB also provides other word sense disambiguation methods besides Personalized PageRank. Even though their optimization is not a focus, their performance could also be observed by applying the same changes made to Personalized PageRank.

There are a few general ways to improve performance of PPR. The first is to manipulate the input given to PPR, which means changing the contextual information given to PPR. Another way to improve performance is to adjust the options of PPR, such as the damping factor and stopping threshold. A final approach is to adjust the source code itself, such as changing the way PPR applies initial weights.

For the manipulation of input, future tests will closely resemble the previous ones. However, they will be more thorough because they will incorporate tagged data. The manipulation of input will still involve adding in WordNet relatives. Other methods could be devised which remove contextual words that are too distant from the target verb. A more efficient metric will be developed to analyze the results quickly so more tests can be done, in comparison to the mere five tests already done. The adjustments of PPR options will definitely look more deeply into using different damping factors. Other options that will be explored are different numbers of PageRank iterations, and different stopping thresholds. Finally, adjustments to the source code could involve any number of things. Currently I would like to focus on having the program apply a higher weight to the first sense of a verb. High weights could even be applied to the first couple of sense for a verb, because the first few senses are always the most common.

Once the adjustments are made and tested, then the best adjustments can be selected and combined. The best performing contextual input format, damping factor, stopping threshold, and first sense weighs will be combined together to achieve higher verb sense disambiguation performance. This approach is critically dependent on the higher weights of first senses. If that implementation is successful, then performance rates could be above 80%, considering that the first sense method performed at about 77%. PPR performance above 80% would be exciting because PPR baseline performance was only 43.8% for this project. Agirre and Soroa's PPR performance using WordNet 3.0 lexical knowledge base was only 41.5%.

References

[1] R. Mihalcea, P. Tarau, and E. Figa, "Pagerank on semantic networks, with application to word sense disambiguation," in *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2004, p. 1126.

[2] S. Banerjee and T. Pedersen, "An adapted lesk algorithm for word sense disambiguation using wordnet," in *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*. London, UK: Springer-Verlag, 2002, pp. 136–145.

[3] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 33–41.
[4] A. Giuglea and A. Moschitti, "Knowledge discovering using framenet, verbnet and propbank," p. 6, 2004.
[5] M. T. Pazienza, M. Pennacchiotti, F. M. Zanzotto, and V. B. Arcimboldi, "Mixing wordnet, verbnet and propbank for studying verb relations."
[6] [Online]. Available: http://framenet.icsi.berkeley.edu
[7] [Online]. Available: http://wordnet.princeton.edu/wordnet
[8] [Online]. Available: http://ixa2.si.ehu.es/ukb
[9] [Online]. Available: http://nlp.stanford.edu/software/tagger.shtml
[10] E. Agirre, O. L. D. Lacalle, and A. Soroa, "Knowledge-based wsd on specific domains: Performing better than generic supervised wsd."
[11] A. Burchardt, K. Erk, A. Frank, A. Burchardt, K. Erk, and A. Frank, "Lecture a wordnet detour to framenet."
[12] R. Johansson and P. Nugues, "Using wordnet to extend framenet coverage," *In Proceedings of the Workshop on Building Frame-semantic Resources for Scandinavian and Baltic Languages, at NODALIDA, Tartu, Estonia, May 24*, 2007.
[13] [Online]. Available: http://www.senseval.org