# Generating Giant Word Corpora with Human Computation to Solve Word Sense Disambiguation

Jonathan Chu
Undergraduate Senior Thesis
Advisors: Dr. Anthony Tomasic, Dr. Luis von Ahn

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

December 3, 2010

**Abstract**

Given a word within a sentence, how can a computer determine the meaning of that word? If there is only one given definition of the word, the solution can be easily determined. If multiple definitions exist however, this problem becomes magnitudes more difficult. The open problem of Word Sense Disambiguation, hereafter referred to as WSD, has yet to be adequately solved and is considered an AI-Complete problem. The applicability of machine learning to this problem is obvious. However, the major issue with such an approach is a lack of data with which we can train a machine learning algorithm. Our solution to this issue is to apply Human Computation. We will create a game that will generate a giant corpus of (word, sentence) pairs tagged with a disambiguated definition for that word from previously untagged sentences. Using this corpus, we will gain the ability to successfully train a machine learning algorithm to be robust enough to solve WSD on a domain as large as an entire language. Thus, an effective solution to WSD can be created...

# 1 Introduction

There are thousands of search queries executed every second in the technical world. Information seekers from the United States to China are leveraging technologies like Google.com and Bing.com to find the information they seek on the web. On a smaller scale level, organizations such as companies and universities provide their own search services to their employees to find things as simple as a phone number or as complex as an negotiation report throughout their own internal data. In the age of technology, search is a critical tool. Currently, however, a computer processing a search query cannot accurately determine the meaning of the query itself when deciding what determines a relevant search result. Thus, the quality of search suffers. In machine translation, the problem of semantic meaning also arises. How can you accurately translate a phrase if the meaning of the phrase is not known? This problem of determining the semantic meaning of a word within a given context is known as the problem of Word Sense Disambiguation. Once solved, Word Sense Disambiguation can be applied to many current problems including text based search, and natural language processing. By applying semantic search to keyword search, it is theorized that the accuracy of search can be significantly increased, both revolutionizing and improving current technology.

**Outline** The remainder of this article is organized as follows. Section 2 defines the problem we are attempting to solve. Our new approach to this problem is detailed in Section 3. The Sections 4 and 5 detail how we generate training data from our game. In Sections 6, 7 and we run a preliminary version of Jinx and detail our results. Finally, in Section 8, we detail our conclusions from this experiment.

# 2 Defining Word Sense Disambiguation

WSD is an open problem in natural language processing formally defined as the process of determining the semantics of a word within a given context. It is an AI complete problem that was first defined as a computational task in the 1940's making it one of the oldest problems in computational linguistics [6]. When attempting to solve WSD, in the trivial case where a word has only one given definition, the process of WSD is simple for it reduces to merely

searching for that single definition within an electronic dictionary. When a word, however, has multiple senses, the process of determining its meaning becomes non-trivial. For example, given the sentence "I went to the bank to deposit money," a human can determine that the word "bank" is informally defined as a financial institution within that context. A computer, however, cannot easily determine that the definition river bank was not meant instead [5].

Word Sense Disambiguation is usually split into two categories: course grained and fine grained WSD. In the course grained problems, words are only disambiguated at the homograph level. Distinction between semantic meaning is made, but little else is determined. On fine grained challenges, much more extreme distinctions may be made, such as part of speech [9].

# 3 Improving the Supervised Machine Learning Approach

An extremely large variety of techniques for WSD have been researched, but one approach that shows promise is a supervised machine learning approach. In this approach, a classifier is trained for each distinct word by a corpus of manually dictionary tagged and disambiguated examples of that distinct word. It has been shown that on coarser grained WSD problems, disambiguation accuracy can reach above 90%. In finer grained disambiguation, accuracy generally ranges from 50-70%. In comparison, a project known as Senseval/Semeval that aims to measure the accuracy of disambiguation systems has found that humans can disambiguate coarse grained precision with 97% accuracy and fine grained precision with 96.5% accuracy [4]. One of the major issues with using a supervised machine learning, however, is that it necessitates a manually tagged corpus of training examples [13]. Today, a training corpus for something as vast as an entire language is not readily available and would be expensive to generate. Thus, this currently renders supervised machine learning as an ineffective means for solving this problem.

To remedy this issue, we have developed a new approach to generating this training corpus in order to allow supervised machine learning to be applied to WSD on a language as a whole. We apply the notion of Human

Computation to WSD by creating a game that will leverage the brain cycles of humans to generate tags on sentences in order to create a full corpus of the entire English language. This game, which we have deemed Jinx, follows an approach similar to the ESP game first created by Dr. Luis von Ahn and Dr. Laura Dabbish of Carnegie Mellon [12].

# 4 Jinx. The Game.

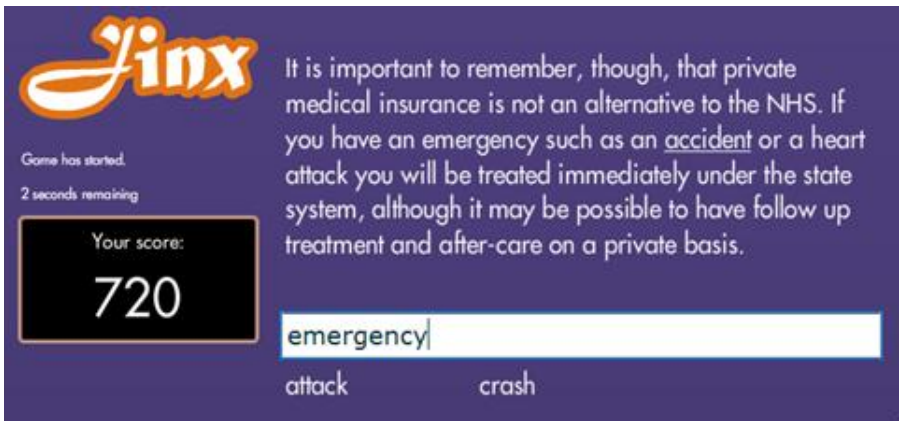


Figure 1. Screenshot of Jinx round in progress.

At the beginning of a round in Jinx, two players will be presented with a context, usually a sentence, and an underlined word to be disambiguated within that context, as shown in Figure 1. The objective of the game is for the two players to enter synonyms for the underlined word and attempt to match each other in order to increase their points. In practice, we have found that players tend to submit words that are in the general category of the underlined word in order to make a match.

Additionally, players are frequently cycled with other players in order to ensure the highest level of randomization possible. This anonymous randomization of players is used to prevent cases of player cheating, for a player does not know who he will be matched with prior to the round beginning. Thus, he is unable to collaborate with that other player and fix the answers.

# 5 Creating a Corpus from Jinx

After Jinx is played, the data that we ultimately receive will be words within a context associated with synonym sets. Thus, consider the following example sentence and hypothetical player matches.

**"I went to the bank to deposit money."**
**Matches: [building, company, finance, money]**

From the output generated by a round of Jinx, we now have a tuple *(bank, I went to the banks to deposit money, [building, company, finance, money])* in the form *(underlined word, context, [synonym list])*. This in itself is not enough to provide us with a training corpus from which a supervised machine learning algorithm can be trained. From here, however, we can create a training corpus ourselves.

Currently, there are many dictionaries that provide synonyms for words as well as the definition. The premier lexical database of English is a well known project hosted at Princeton University called WordNet. This database contains what is to be considered as the gold standard or accepted definitions for semantics in the English language [3]. Associated with each semantic definition for a word on WordNet is a set of synonyms relevant to that word and an example sentence or two [2, 8, 7]. Thus, WordNet closes the final loop that allows us to map a word and a context to a semantic definition. By mapping the synonym sets we obtain from Jinx to the synonym sets present on WordNet, we can identify a semantic definition for a word and thus, disambiguate that word. We may be able to gain even more accurate results by using the example sentences provided by WordNet. By themselves, those example sentences are too few to train a reliable machine learning algorithm [10]. If we run Jinx on those example sentences, however, we obtain a user generated synonym set for that semantic meaning. Thus, the likelihood that the generated synonym sets for an untagged sentence would match WordNet's increases. Therefore, through Jinx, we have the potential to generate a training corpus that can encompass all of the English language.

**Generating Definitions** Thus, for any given word in a context, we generate a definition using Jinx and WordNet with the following steps:

1. Take our word and context and feed it to Jinx, resulting in a unique synonym set identifying the definition.
2. Compare the synonym set to known synonym sets in WordNet.
3. Find the synonym set in WordNet that most closely resembles that of the generated synonym set.
4. Return the definition associated with the synonym set in WordNet.

# 6 Testing a Proof of Concept and Validating Assumptions

In theory, our synonym mappings work out beautifully, but in practice, there are many unknowns that could push a system towards failure. Thus, to validate that our system can correctly tag sentences with a semantic definition, we must first compare its output to a control data set that is already established to be correct. Senseval/Semeval is a competition that is designed to test the accuracy of word sense disambiguation schemes [4]. There has already been much prior research effort put into constructing the Senseval/Semeval competitions to provide accurate metrics to gauge the performance of WSD algorithms. Thus, it is only natural that we use Senseval/Semeval's data as our testing grounds.

To test our system, we recreate the WordNet situation, but utilize Senseval/Semeval's data instead. The Senseval/Semeval corpus that we utilize is the Hector database compiled by the University of Brighton and the Oxford University Press and was the initial database utilized to perform studies of Senseval/Semeval's validity of a WSD system scorer [1]. Senseval/Semeval provides us with three data sets: a test file, a train file, and a gold standard file. The test file has words marked within sentences that need to be disambiguated. The training file has words within sentences that are already tagged with definitions. The gold standard file has accepted definitions for the sentences in the test set generated by a human lexical analyst [3].

We begin by first using Jinx to generate a list of synonyms for all words in the training data. Because this the training data is associated with a definition, this generates a data set to which we can compare out test data. The

generated tagging effectively replaces WordNet. We also run Jinx on the test sentences and then match the synonym sets generated from the test sentences to those generated by the training sentences. Once this step is complete, we have effectively tagged out sentences in the test data with definitions and have created more sentences that we can use to train a supervised machine learning algorithm. To show that we accurately label the test data with the correct definitions, we reference the gold standard data to ensure that we correctly label the test data.

# 7 Accuracy of Tagged Data

More refinement to this model of data generation for Word Sense Disambiguation is still needed in order to fully realize the potential of this method. However, preliminary trials with students playing Jinx appear to be promising.

It is noteworthy that the results of the synonym sets generated by Jinx are common lexical synonyms used in everyday English language. These common synonyms do not always find matches with the ones listed in WordNet 3.0's synonym sets. An example of this was the word "bitter." In common speech, the word "bad" can generally be thought of as a common replacement for the word "bitter," as evidenced by an extremely high number of players choosing the word "bad" as a synonym. However, WordNet 3.0 does not include "bad" as a viable synonym for the word "bitter" [2].

In order to measure the viability of the generated data, we labeled every tagged word from our trials as either identifiable, unmatched, missing, partial, or ambiguous. Identifiable words are words whose synonym sets uniquely matched with a synonym set for a specific semantic definition in WordNet. Unmatched words are words for which no synonym set could be matched. Missing words are words were not present in the WordNet database. Partial matches are words whose elements in its synonym set had partial matches to WordNet's synonym sets (e.g. orchestra matching dance orchestra), and ambiguous are words where this method failed to disambiguate the word for it matched multiply synonym sets.

| Label | Number | Percentage |
|---|---|---|
| Identifiable | 45 | 54% |
| Unmatched | 24 | 29% |
| Missing | 2 | 2% |
| Partial | 3 | 4% |
| Ambiguous | 9 | 11% |

Table 1. Result Summary.

As we can see in Table 1, the data shows that Jinx is extremely promising in creating training data. 54% of the words were identifiable, and 4% of the data had a partial match. Therefore, in total, the first iteration of Jinx was able to provide 58% of the words in our trial with valid definitions and can potentially create valid training data for these words.

However, 29% of the words were unmatched and 11% ambiguous. Finally, 2% of the words were missing from WordNet [11]

In future iterations of the game, the idea of taboo words from the ESP game can be borrowed in order to improve results [12]. These words are illegal words that cannot be entered by a player to gain points. By doing so, we will be able to increase the variety of synonyms we obtain for each round and thus, should be able to significantly reduce the number of unmatched and ambiguous words. For the 2% of words missing from WordNet, we can only hope for future versions of WordNet to be more comprehensive, or for better linguistic resources to surface.

Additionally, it is important to note that identified and partial words does not necessarily mean that the associated word sense is the correct sense. Of the 48 identified and partial words, 14 of them had very limited synonym sets that provided no ability to disambiguate. These would greatly benefit from the use of taboo words.

# 8 Conclusions

Throughout our discussion, we have identified and explored a new technique for generating data to train machine learning algorithms. We specifi-

cally tackled the problem of Word Sense Disambiguation by applying novel techniques in Human Computation.

Using humans to provide us with data allows for a cheap and efficient way to produce high quality data that can be used for machine learning training sets. The way in which we are able to capture human brain cycles is by capturing their interest through a fun and enjoyable game named Jinx. After running a test trial, participant feedback indicated that Jinx was an enjoyable game capable of sustaining large audiences to perform our computations.

Data from our test trial with our first iteration of Jinx shows promising results. The game has the ability to disambiguate sentences and create viable training data for algorithms. However, although much potential has been shown, this method of data generation still has many major issues. In our trial it was unable to disambiguate a non-trivial number of words and unable to provide any useful data for those words as a result. With further research and iteration, it is possible that Jinx will only fail for a small number of words and provide high quality data for a vast majority of the English language.

# References


[1] S. Atkins. Tools for corpus-aided lexicography: the hector project. *Acta Linguistica Hungarica*, pages 5–72, 1992-1993.

[2] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[3] A. Kilgarriff. Gold standard datasets for evaluating word sense disambiguation programs. *Computer Speech and Language*, 12(4):453–472, 1998.

[4] A. Kilgarriff. Senseval: An exercies in evaluating word sense disambiguation programs. *LREC*, pages 581–588, 1998.

[5] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. *SIGDOC-86: 5th International Conference on Systems Documentation*, pages 24–26, 1986.

[6] R. Mihalcea. *Encyclopedia of Machine Learning: Word sense disambiguation*. Springer, 2007.

[7] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[8] G. A. Miller. Wordnet - about us, 2009.

[9] Navilgi, Litkowski, and Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. *Semeval-2007 Workshop (SEMEVAL), in the 45th Annual Meeting of the Association for Computational Linguistics*, pages 30–35, 2007.

[10] P. Resnik and D. Yarowsky. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133, 2000.

[11] N. Seemakurty, J. Chu, L. von Ahn, and A. Tomasic. Word sense disambiguation via human computation. *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP)*, 2010.

[12] L. von Ahn and L. Dabbish. Labeling images with a computer game. *ACM Conference on Human Factors in Computing Systems*, pages 319–326, CHI 2004.