

Human-Powered Word Sense Disambiguation

Nitin Seemakurty, Jon Chu, Luis von Ahn, Anthony Tomasic

School of Computer Science

Carnegie Mellon University

Pittsburgh, PA, USA

{nseemaku, jechu, biglou, tomasic}@andrew.cmu.edu

ABSTRACT

One formidable problem in language technology is disambiguating the true sense of a word as it occurs in a sentence (e.g., recognizing whether the word "bank" refers to a river bank or to a financial institution). This work explores a specific strategy for solving this problem. The strategy involves harnessing the linguistic abilities of human beings to develop datasets that can be used to train machine learning algorithms. Generation of quality datasets can greatly aid the development of algorithms that tackle challenges such as automated language translation and the development of a semantic web. To create such datasets, we introduce a new interactive system: a fun game designed to produce valuable output by engaging human players in what they perceive to be the casual task of guessing the same word as another player. Our system makes a valuable contribution by tackling a bottleneck in the WSD domain: knowledge acquisition. Rather than using conventional and costly techniques of paying workers to generate training data for machine learning algorithms, we delegate the work to people who are looking to be entertained.

Keywords

Word sense disambiguation, sense tagging, distributed knowledge acquisition, games with a purpose, online games

INTRODUCTION

The human language is ambiguous. That is, words can be interpreted with different meaning depending on their surrounding context. Take, for example, the following two sentences [7]:

- (a) I can hear *bass* sounds.
- (b) They like grilled *bass*.

The word *bass* refers to a low-frequency tone in one sentence while it refers to a type of fish in the other. Although this sense recognition seems intuitive to humans, it is a much more sophisticated task for a machine, which has to cope with the unstructured nature of the data (language). This computational identification a word's meaning in a given context is called *Word Sense Disambiguation (WSD)*.

The relevance of WSD is becoming clear as advancing information/web technologies are catalyzing the production of enormous amounts of textual data, including articles, blogs, status messages, digitized books, etc. There is a growing need to introduce structure to this data in order to

make it consumable and manageable by machines. Unfortunately, WSD remains a difficult problem for several reasons.

Most relevant to our study is the issue of knowledge acquisition. At its core, a WSD system is one that utilizes available sources of knowledge to calculate the most relevant meaning of a given word in context. But manually creating a training dataset for a WSD system generally involves taking a large set of textual data, isolating words to disambiguate, and hand labeling each of these words with their "correct" meanings. We quickly find that this process is an arduous and consequently expensive one [6].

But what if we make this labeling process a pleasant one? This paper explores a new system, a game that is designed to capture human knowledge in a distributed fashion via an enjoyable game. Our study involves *assessing the effectiveness of this game in tackling the knowledge acquisition bottleneck*. Many elements of our system are derived from a predecessor: the ESP Game [10].

Open Mind Initiative

Like the ESP Game, our game is much in tune with the efforts of the Open Mind Initiative [9], which focuses on collecting data from internet users in order to train machine learning algorithms. Our game is similar in that it attempts to use the efforts of regular internet users to tag the senses of words. However, as with the ESP Game, we place particular emphasis on the playability (i.e. survivability) of our system.

GENERAL GAME PLAY

Jinx is an online two player game. When a player begins the game, he/she is paired with another random player. The player does not know who his/her partner is and the game does not facilitate any form of communication between the two players. Each player interacts with the game independently. The players share only one aspect of the game: the current round. At any given time, both players view the same round, where a round is defined by a context (e.g. a sentence), and a highlighted word within that context.

The players are encouraged to rapidly type replacement words/phrases for the highlighted term. They are given incentive to type words that their partner is likely to type because both players are awarded points if and only if they both type the same string. As with the ESP Game, these

players do not need to type their matching string at the exact same time, but both must have independently typed this string at some point during that round (See Figure 1 below).



Figure 1. Each player guesses word replacements independently. Neither player can see the other player’s guesses.

We call this matching string a “tag”. Once a tag is collected, the game awards points to each player and then proceeds to the next round. In the case where agreement cannot be reached, the round expires after 30 seconds. Players are presented rounds for exactly 3 minutes, and then they are taken to a summary page that recaps their performance and offers to restart the game.

Our observations currently indicate that these collected tags are typically appropriate synonyms for the highlighted term.

GAME DESIGN DETAILS

Originally, our game was designed to be more of a quiz comprised of a series of multiple choice questions. The player would be presented with a highlighted word in context, and then given multiple definitions to choose from. These definitions were intended to reflect different interpretations of the highlighted term. The player would be rewarded if his/her choice matched with the partner’s choice. This setup, however, was inherently flawed.

- (a) **Random guessing.** This older setup allowed players to collect occasional points by blindly selecting answers and rapidly progressing through rounds, hoping for a lucky match with the partner. One solution could be to penalize for mismatch, but that would mean that a normal player would be deducted points due to the misbehavior of his/her partner. Our solution to this problem was inspired by looking at yet another flaw with this setup.
- (b) **Rigidity.** Limiting answer choices to a set of dictionary definitions also had the potential to create confusion if none of the definitions “worked”. Players would eventually choose an answer that only weakly approximates the word’s true meaning. From our perspective, this translated to weaker data.

- (c) **Playability.** Like the ESP Game, we diverge from the work of Open Mind by emphasizing the element of fun in human knowledge contribution. A game that demands a player to read through dictionary definitions and merely click through rounds would quickly be abandoned by the online community.

For these reasons, we adopted a more open-ended approach that allows players to rapidly guess word replacement strings. This approach minimizes a player’s ability to match by randomly guessing. At the same time it makes the game more challenging and engaging by requiring that the players guess cooperatively, despite not being able to communicate. This cooperation emerges automatically from the task at hand and results in the generation of valuable tags.

Tag Quality

Because there are only a few word replacements that are relevant to any given round, players quickly recognize that making guesses from this limited set drastically increases their chances of matching with their partner. Consequently, the tags collected from the game are typically relevant word replacements.

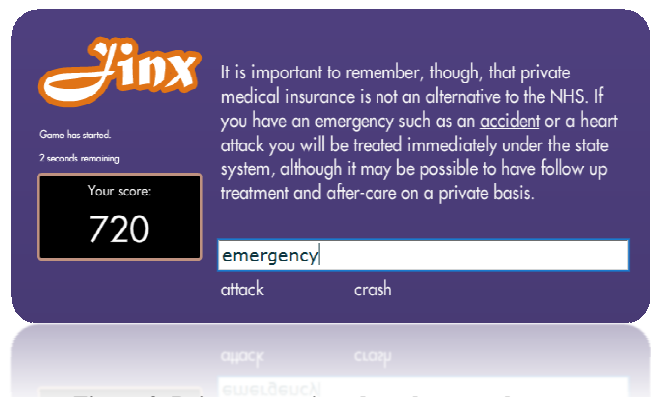


Figure 2. Points are assigned to players only on a match. The number of points rewarded depends on several different factors.

Point system

A game’s reward system can drastically affect player behavior during game play. In designing the point system for Jinx, we had several goals in mind. We wanted to keep the game fast paced while still allowing for high quality input from players. Fast pace is encouraged because prior dry runs of the game indicate that the matching tag (i.e. the best replacement for the word) is commonly a very early guess during the round (one made quickly after reading the provided textual context). Giving players a sense of urgency encourages them to guess what is most intuitive to them, and this tends to be a successful tag. To generate this urgency, upon a matching guess, we reward each player $P = f(t)$ points, where t is the number of seconds remaining in the round when the matching guess was made, and f is an increasing function (we currently use $f(t) = 10 * t$). The faster a player generates a tag, the more points he/she makes. Notice that one of the players will inevitably

make the matching guess before the other player does; each is rewarded accordingly.

Players are also awarded bonus points for successfully matching with their partner on consecutive rounds. The value of the bonus increases on every round of their consecutive streak. At the end of a round, we award each player $B = g(n)$ bonus points, where n is the number of consecutive matches so far, and $g(n) = 10 * n$. Note that while g is a linear function, its effect is geometric because the total bonus a player receives is $10 * (1 + 2 + 3 + \dots)$. This bonus system not only encourages players to keep playing, but also to keep playing with accuracy in mind. Most importantly, it keeps the game exciting.

Bots

During the course of a round, our game records all of a player's guesses for that given highlighted term in context, along with the guesses' timestamps. Whenever we need to deploy a bot, we intend to simply "replay" these guesses with the same delays as when originally recorded, as if they was coming from a human player. The partner to a bot should be completely unaware of its automation; all guessing and matching proceeds normally.

Because our game is a two player game, it is always possible for one player to have no one to pair with. In such a scenario, a bot will be equipped with a recording of a similar previous round and then paired with the solo player.

Bots will also be used to populate the game so that random matching can continue reliably. We discuss the importance of such matching in preventing cheating below.

Cheating

To maintain tag quality, it is imperative that paired players in our game are unaware of each other's identity. Otherwise, their guesses can easily be coordinated to match and then produce a bogus tag. While our system is designed to account for occasional bad tags, communication between two paired players can result in bad data.

It is easy for this communication to happen if both paired game instances are running at the same location. In such a case, there is only one player controlling both instances and it is trivial for that player to match on guesses. To minimize the chances of this one-player scenario, we record IP addresses and ensure that paired game instances run at separate IPs. It is also possible for a player to spawn multiple instances of the game from separate IPs in rapid succession, hoping that at least two instances under his/her control are paired. To mitigate the threat of this scenario, our server pairs all waiting players only once every thirty seconds (inspired by the ESP Game). This ensures (with high probability) that more than one player is in the waiting pool at the time of pairing.

Another threat to the game is mass-strategy. Mass-strategy refers to a consensus among a significantly large group of players to coordinate all their input (e.g. always make the same guess). The Internet makes the organization of such

an attack very feasible. Inspired by the ESP Game, we aim to design our game such that when a mass-strategy is detected, a large number of bots can be immediately deployed. The pre-recorded game play of these robots severely limits the effectiveness of the mass-strategy. We presume that this bot infusion will be enough to render the mass-strategy a useless endeavor.

USING COLLECTED DATA

To evaluate the game's effectiveness in collecting quality tags, we have injected testing data provided by the HECTOR project [1] into our system. This project provides us with a significant set of contexts (sentences) with "highlighted" terms, as well as the definition that best suits that word in the given context. Using Jinx, we aim to collect tags for the highlighted terms exclusively in that dataset, and then verify that these tags correspond to correct definition of the term.

WordNet

We later intend to use Princeton University's WordNet [5] to map the tags we have collected to machine readable, standardized definitions. We plan to do this by utilizing the "synsets" provided by WordNet. Synsets are essentially a group of synonyms that, for the purposes of information retrieval, are semantically equivalent. WordNet attributes a simple but representative definition to each synset. By recognizing which WordNet synset most intersects the set of tags collected for a word, we can link that word to the synset's affiliated definition.

CONCLUSION

Ultimately, we hope to build a high quality dataset comprised of the following mapping:

$$(word, context) \rightarrow definition$$

We aim to do this in a distributed fashion by employing human beings to take a word in context and generate its tags, tags which later serve as links to the word's correct definition. The total mapping we generate can then be used as training data for machine learning algorithms.

To facilitate this process, we introduce our new system (Jinx), which is designed to effectively harness the linguistic abilities of human beings so that this mapping can be generated with quality and cost-effectiveness.

We hope that the data collected from Jinx can significantly lower the barrier to entry in WSD study and expedite, if not enable, the development of novel algorithms for language technologies.

REFERENCES

1. A. Kilgarriff. 1998. Senseval: An exercise in evaluating word sense disambiguation programs.
2. Anderson, R.E. Social impacts of computing: Codes of professional ethics. *Social Science Computing Review* 10, 2 (1992), 453-469.

3. How to Classify Works Using ACM's Computing Classification System.
http://www.acm.org/class/how_to_use.html.
4. Mather, B.D. Making up titles for conference papers. *Ext. Abstracts CHI 2000*, ACM Press (2000), 1-2.
5. Miller, G. A. 1995. WORDNET: A Lexical Database for English. *Communications of ACM*
6. NG, T. H. 1997. Getting serious about word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?* (Washington D.C.). 1-7.
7. Roberto Navigli. 2009. Word sense disambiguation: a survey. *ACM Computing Surveys*, 41
8. Schwartz, M. *Guidelines for Bias-Free Writing*. Indiana University Press, Bloomington, IN, USA, 1995.
9. Stork, D. G. The Open Mind Initiative. *IEEE Intelligent Systems & Their Applications*, 14-3, 1999, pages 19-20.
10. Von Ahn, L. and Dabbish, L. 2004. Labeling images with a computer game. In *Proc. ACM CHI.NG 1997 Paper*
11. Zellweger, P.T., Bouvin, N.O., Jehøj, H., and Mackinlay, J.D. Fluid Annotations in an Open World. *Proc. Hypertext 2001*, ACM Press (2001), 9-18.