A Real-time N-gram Approach to Choosing Synonyms Bases on Context

Abstract. Synonymy is an important part of all natural language but not all synonyms are created equal. Because two words are synonymous doesn't necessarily mean that they can always be interchanged. The problem that we address is that of near-synonymy and choosing the appropriate near-synonyms based purely on the surrounding words. Our computational method, unlike previous methods developed for this problem, is capable of making multiple word suggestions, which more accurately models human choice, it contains a large number of words, it does not require training, and it runs in real-time. When able to make multiple suggestions on testing data developed for the early studies of near-synonymy, it improves by over 17 percentage points on the previous best method and 4.5 percentage points on the human annotators' near-synonym choices. This paper also presents new synonym sets and human annotated test data that better represents the problem that we are studying.

1 Introduction

Near-synonymy, often simply referred to as synonymy, is the closeness of meaning among words. Near-synonymous words can sometimes be substituted for each other in sentences. The problem that we attempt to address is that of context-sensitive near-synonymy: the choice of a possible near-synonym substitution based purely on its surrounding words. This is usually not an easy problem, because natural language, specifically English, is such a varied and unconstrained language that there are no hard and fast rules that govern all word usage. For these sentences *He found it [difficult] to believe.* and *He found it [hard] to believe.* which near-synonym do we choose? What is more accurate given this context *difficult* or *hard*? The choice is not always an easy one and it could be argued that both words are fitting here. But when we use another near-synonym we begin to see that there are times when context does prevent the use of some words, *He found it [demanding] to believe.* for instance. Synonyms are actually a group of words that can be divided into two sets, absolute and near.

Absolute synonymy means that two words share exactly the same meaning and all the same nuances. In theory an absolute synonym would be completely interchangeable with any other synonym in any context. But, even in an optimistic light, absolute synonymy is very rare.

Near-synonymy is usually what most of us think of when we hear the word synonym and this is the word relationship that we focus on in this paper. Near-synonyms are words which are close in meaning. They are close to being absolute synonyms but are not exactly the same. They are very similar in meanings but have different usage patterns and convey slightly different nuances or meaning [5]. It is near-synonyms which make up a common thesaurus. For example the words *error*, *mistake*, and *blunder* are all near-synonyms to each other but each has slightly different connotations and can't always be used in the same context. **Synonym** is used to mean **near-synonym** in the remainder of this paper when used in contexts in which no confusion could arise.

The motivation behind our new method is to use the vast amount of the new data available in the form of Google n-grams to create a fast and practical method that is able to find a proper near-synonym that fits a particular context by attempting to match the surrounding words in an idiomatic fashion with similar ones in the n-gram data. Although other foundational work has also used Google n-grams to suggest near-synonyms [6], to our knowledge our system is the first capable of making multiple word suggestions and the first capable of running in real-time. Previous methods that were capable of making only a single word suggestion for each synonym set were limited. There is very rarely a context where only one near-synonym is possible and by restricting testing to only one word has made this problem one which is trying to find only the word specified by a particular human writer rather than finding a set of possible synonyms which fit the context which is our aim.

The contributions presented in this paper are as follows:

- This new method, unlike previous methods used on this problem, is capable of making multiple word suggestions which more accurately models human choice.
- This system is capable of running in real-time on a standard computer. This could allow for the use of our system in many future user specific applications, including word processing software.
- New larger extensive synonym sets and testing data are introduced which we believe are more suited to testing this problem.

2 Related Work

This section surveys the foundational research papers that have dealt with the rather elusive problem of near-synonymy and choosing the right near-synonymous word in different contexts.

Philip Edmonds' paper [2] was amongst the first to tackle the problem of near-synonymy by using a large corpus combined with an idiomatic approach. Edmonds notes that knowledge-based approaches to this problem in the past had suffered from "serious lexical acquisition bottlenecks" that apparently stemmed from human annotation. In his paper he presents a new statistical approach to solving a problem of choosing "the most typical" word from a small list of possibilities (usually only 3) to fill a lexical gap created by removing the

original word. This fill-in-the blank method is a straightforward but somewhat problematic way of evaluating this and similar methods. The words used in the selected set of near-synonyms are chosen to be of low polysemy in an attempt to minimize the chance of a word having multiple senses. Table 1 shows Edmonds' 7 near-synonym sets. As noted this approach to evaluation is not a perfect way of testing this because only the near-synonyms chosen by the authors of the Wall Street Journal (the source of the corpus) are deemed correct and the authors are not always typical and many different synonyms can be used in a given context. This evaluation also worked well in this instance because the purpose of this paper was to determine the "most typical" synonym used in a specific context rather than finding the best. Though despite any short-comings this evaluation and these sets have, they have endured and continue to be used in the majority of subsequent papers dealing with this problem of structural differences in near-synonymy.

Table 1. Edmonds' 7 sets of near-synonyms

Set	Synonyms
1	Difficult, hard, tough
2	Error, mistake, oversight
3	Job, task, duty
4	Responsibility, commitment, obligation, burden
5	Material, stuff, substance
6	Give, provide, offer
7	Settle, resolve

The method presented in this paper builds a lexical co-occurrence network and uses second order co-occurrence relations in order to choose the best word. By using second order co-occurrence relationships, results can be uncovered without a massive volume of text. A very large corpus would be required when simply looking at direct first order co-occurrence. This method does not need such a large amount of data but does require that a lexical network be built in advance for every group of near-synonyms used. Even though words may never occur together, these second order co-occurrence relationships might show that the occurrence of the first word may be used to predict that the other word will be used. Edmonds lexical co-occurrence network is built layer by layer. A root word is connected to other words which have significant co-occurrence and this process is recursively repeated for each word. A first-order co-occurrence is determined by the intersection of two well-known measures of significance, mutual information scores and T scores [2].

Nearly 10 years after Edmonds first took on this task of examining context for near-synonymy, Diana Inkpen [4] addressed the same problem but used a much larger corpus with simpler methods and was able to obtain improved results. The Waterloo multi-text system was used as a corpus. It contains about 1 TB of texts collected by a web crawler. Inkpen's first method computes a score

for each candidate near-synonym to fill the space created by removing one of the near-synonyms from a sentence. The score is based on point-wise mutual information (PMI) between the candidate and other content words appearing in the context. In this method stopwords are filtered out and do not play a role in candidate selection. The best results in the paper were obtained using small window sizes, $k = 1$ and $k = 2$, and performance quickly declines as the window size is increased. Although not explicitly stated, it is fair to assume that this method is relatively slow to compute because it remotely queries the Waterloo terabyte corpus. Inkpen uses the same evaluation techniques presented by Edmonds, but she also addresses some of the inherent problems in those techniques and suggests that the best way to evaluate this algorithm would be to use human readers but notes this kind of evaluation would be incredibly time-consuming. Inkpen even did some limited trials to verify this with human judges on a small sample of the data set, each containing 350 sentences. The results show a high but not perfect agreement of 78.5% which shows this task is indeed difficult and there is no one correct way of doing it. Synonym usage will vary according to writing style and several other factors.

Several years after Inkpen published the previous paper, she co-authored a paper with Islam [6] which presented a new unsupervised statistical method for solving the same near-synonymy problem. This new near-synonymy method uses a new and different type of corpus, the Google Web 1T n-gram data set [1], collected from over 1 trillion webpages. This corpus is composed of unigrams to 5-grams and includes their observed frequency counts. Because of Inkpen's previous results and the fact that the best results were achieved with a small window size, it is natural to assume that this very large corpus, even being limited to a five word scope, would contain enough information to make a decision about near-synonym choice. The method presented in this paper considers up to four words before the lexical gap and up to four words after. This window results in at most 5 queries to the Google Web corpus per near synonym. Queries by this method are done recursively meaning that if no-grams are found to match the current context a smaller n-gram is used from 5-grams down to bigrams. This makes this method rather time-consuming, especially in a real world situation compared to a test with a limited number of synonym sets. This is simply due to the number of queries that are issued and the size of the unmodified Google Web corpus.

3 Proposed Method

The basic premise of our work is to create a small, fast and portable system that can be used in a real-life situation that addresses the problem of contextual near-synonymy. Up until now all the known systems that that have been created to handle this problem are very slow, very large [2] [4] [6], or require some kind of advanced knowledge about the specific writing style the system can be used on [4], and possibly the greatest limitation is that all of them need to be trained

on each synonym group to be used on which negates any prospect of them being used for any non-research purpose.

Our system uses one of the 5-gram datasets of the n-gram datasets released by Google: the ENGLISH ONE MILLION version 20090715 which contains the one million most common words. A 2009 and a 2012 set created from the scanning of published literature exist [3]. These n-grams are fixed blocks of n words collected by Google as it digitized 5.2 million books published between 1500 and 2008. There are several Google n-gram data sets in existence. The 2006 web 1T set [1], compiled from indexed webpages through the Google search engine, was used in [6]. We ran some initial trials on this data set, but the unchecked grammar and often messy html tags did not seem to offer the results desired by our system. Its size was also a limiting factor, as well. The decision to use this smaller subset was influenced by the goal of making the program run quickly (in real-time) and also the time needed to construct the database. This smaller subset required the Cognitive Engineering Laboratory system to run continuously for two weeks to construct this database which made using the full 2012 set of 500 billion words unfeasible on the current system. This full 2012 version of the 5-grams is much larger, and would be an interesting set to use in future work.

Unlike the previous methods used for idiomatic matching, our new method does not try to match the entire 5-gram but rather several smaller features found to be important in choosing synonyms based on context. By doing this, it becomes a trade-off between total accuracy and size of the database. To achieve perfect accuracy it would potentially require a very large database that contains every possible English five word context, but instead of trying to achieve this, we focus on creating a practical system.

Perhaps our system's greatest advantage is the fact that it contains over 170 thousand distinct words and does not need to be trained on any specific synonym set prior to use and coupled with its real-time speed allows our system to be used in a viable practical application. More details of our system can be found in [citation removed].

4 Database

The Google 5-grams used by this program are available in .csv format from Google. The 5-grams are converted from their default .csv format into a relational MySQL database to both reduce size and allow real-time access. Uncompressed, these files equal 254 GB but are reduced by over 99% through pruning and conversion to a relational database. The database uses word indexing to allow each string to only be saved onto the disk once regardless of how many times it appears in all of the 5-grams [7].

Extensive pruning was done during database construction in an attempt to clean up the data and remove non-English symbols and other tokens that would either not occur in natural text or was altered by the digitizing done by Google books. For example periods and commas, and usually quotation marks are all considered separate words in the n-grams and because it was felt that grams

containing these would likely contain context from multiple sentences/thoughts, these grams were removed. Every word was first converted to lowercase then only 5-grams in which each gram contained at least one alphabetic character were used.

Four individual feature tables were created to further reduce the size and allow portability. These four features, which are described in the next section, are the before bigram, after bigram, split bigram and important word. The four corresponding tables are approximately 2.4 GB combined.

5 Features

There are four features that are extracted from the 5-grams in the database. These features have been identified by examining the frequency counts in the database to find distinctive contexts that commonly differentiate between near-synonyms. An example of this is the before bigram *the most* which alone eliminates most of the wrong near-synonyms in the original *difficult, hard, tough* synonym set. By using four features from each 5-gram we are increasing the virtual size of the database without adding more grams and increasing the true size. This facilitates real-time operation and minimal database size.

The database is constructed so that two preceding context words as well as two trailing context words are associated with the desired word used in synonym lookup located at position three.

Fig. 1. Before bigram feature

The before bigram feature (bb), visualized in Figure 1, consists of the two context words that precede the desired word. Unlike many previous methods stopwords are also considered. The after bigram feature (ba), visualized in Figure

Fig. 2. After bigram feature

2, uses the two words that immediately follow the desired word. This feature is similar to the previous before bigram feature.

The split bigram feature (bs), visualized in Figure 3, considers the word directly to the right and the left of the desired word. This feature is the highest performing feature in many situations and has one of the highest accuracies among all the features on average.

Fig. 3. Split bigram feature

This performance is particularly notable because it considers stopwords unlike the majority of the previous methods which remove them. In many situations the immediately preceding and following words are often stopwords, and this demonstrates how stopwords can be quite important in this application of choosing among near-synonyms.

Fig. 4. Important word feature

The important word feature (bis) 4 ignores a small list of 41 stopwords and tries to find a word deemed to be a context word in the 5-gram that is within two words of the word in the text being analyzed. The accuracy of this feature seems to be somewhat random. It seems to find correct suggestions that the other features do not but also produces more incorrect suggestions.

6 Results

This section reports on two evaluations of the proposed system for choosing appropriate synonyms. The first evaluation uses data provided by Diana Inkpen. The second evaluation uses data generated specifically for this study.

6.1 Comparison to Previous Research

First, as a means of comparison to previous methods we have evaluated our method on the human annotator data that Inkpen and others have used. We use the seven synonym sets which have become standard for this problem. Although this evaluation is useful to compare this work to the previous research results, the new system's intended purpose and capabilities are different: it is capable of making multiple word suggestions which more accurately models human choice, it contains a much larger number of words that can be used in the synonym suggestions, it does not require training, and it is able to run in real-time.

It is somewhat difficult to make a direct comparison between these previous methods and our system, because unlike them, our system is able to choose more than one adequate answer. Even though the human annotators they used were also able to choose multiple answers when they felt it was appropriate, this option was used sparingly, only about 5% of the time, according to Inkpen.

Table 2. Comparison of results on original synonym sets

Set	Inter-annotator agreement	Inkpen (2007)	Web n-gram method (2010)	Proposed method (one choice)	Proposed method (multiple choices)	Average # of choices	Unknown contexts
[difficult, hard, tough]	72%	53%	62%	69%	86%	2.3	3
[error, mistake, oversight]	82%	68%	70%	59%	72%	1.6	13
[job, task, duty]	86%	78%	80%	59%	88%	2.5	8
[responsibility, burden, obligation, commitment]	76%	66%	76%	53%	85%	3.0	3
[material, stuff, substance]	76%	64%	56%	60%	83%	2.1	15
[give, provide, offer]	78%	52%	52%	44%	83%	2.5	7
[settle, resolve]	80%	77%	66%	57%	84%	1.6	10
AVERAGE	78.5%	65.4%	66%	57.3%	83%	-	-

We have included results for a single answer. With this method, when the program made no suggestion, it chose the first word in the set by default. Comparing these results to the previous ones shows mixed results: some synonym sets saw improvement, while others did not. As indicated in the far right column the number of unknown contexts varied a fair bit throughout the seven synonyms sets and is an indication that the 5-gram database may be lacking in size. It is left for future work to determine how much of an increase in database size would correct this problem.

These tests were run on our system as a means of comparison. Our system unlike previous systems is capable of making multiple word suggestions. It contains a large number of words and is able to be run in real-time. When comparing directly with past systems that can only make one choice, it did not always better these scores when only allowed a single suggestion, but when able to make multiple suggestions it improved over 17 percentage points on average (the correct answer was one of its suggestions) and even improved on the human annotators by 4.5 percentage points on average with a maximum of 14 percentage points. When examining this data these words seem to be far too interchangeable. It is noted in the previous papers that many of these decisions are based purely on writing style of a specific author and even the human annotators they used had relatively low agreement because of this. For example in Inkpen's human annotated data for the set [difficult, hard, tough] the vast majority of the sentences allow for the use of all of these words and accuracy was simply being gauged as the ability to match the word used by one specific human writer in the Wall Street Journal's file.

6.2 New Synonym sets and a different kind of annotation

Historically, to test the performance of solutions to this problem testing has been conducted on the same set of seven synonyms chosen by Edmond in the first paper dealing with this problem with a contextual idiomatic approach [2]. These word sets were reported to be chosen because of their low polysemy, but while this is true on the surface it is not entirely accurate which we believe has led to a testing bias in the past. This is true of the first word in each set but is usually not for every word in the set. This can be seen in the set [difficult, hard, tough]. The word *difficult* does not have many senses but *hard* does. According to Wordnet [8] *difficult* only has 2 distinct senses:

> S: (adj) difficult, hard (not easy; requiring great physical or mental effort to accomplish or comprehend or endure) "a difficult task"; "nesting places on the cliffs are difficult of access"; "difficult times"; "why is it so hard for you to keep a secret?"
>
> S: (adj) unmanageable, difficult (hard to control) "a difficult child"; "an unmanageable situation"

On the other hand, a search for *hard* returns 22 different senses that are spread over Adjectives and Adverbs. This list does not even include it as a part of a noun in such words as *hard drive* which actually appears frequently in the testing data.

This difference in polysemy serves as a major problem in the way testing was previously done. Since testing of each set was performed by removing all occurrences of the words in a set from the testing data and gauging the system's ability to insert the correct word back in, a high polysemy of one word and a low polysemy of another blurs the results by making it artificially easier to distinguish contexts for different words with many senses. It was due to this that it was decided to create expanded richer synonyms sets with new human annotated data that allowed for multiple choices of the word that would fit the context.

To test aspects of the system not exercised previously, two new richer synonyms sets were created, each containing seven synonyms. These sets are given in Figure 5. The first of these sets was created by expanding the already existing set from Edmonds, [difficult, hard, tough], and the other was created from scratch based on two very common and very similar synonyms *little* and *small*. Synonyms were chosen by comparing the synonym lists of the base word from WordNet 3.1 [8] and the Microsoft Word built-in thesaurus.

New human annotated data was collected for these two synonym sets to facilitate a direct comparison with how this method would perform. The sentences were collected in a manner which is believed will eliminate the polysemy that interferes with the previous test. It was decided to use sentences from the same Wall Street Journal 1987 source but only ones that contained one of the two base words from each synonym set, i.e., *difficult* or *little*, to eliminate the possibility of multiple senses for the word. Human annotators judged these sentences to

Difficult	Little
Hard	Small
Tough	Slight
Trying	Tiny
Problematic	Minor
Challenging	Modest
Unmanageable	Insignificant

Fig. 5. The two new synonyms sets created for testing

determine which of the seven words could be used to fill each context. The human annotators were not aware that the original sentences had only *difficult* or *little* in the judgment position. Two human annotators both skilled in English (a BA of English literature and a PhD candidate of English literature) were asked to judge approximately 150 sentences for each set and to choose all the words from each synonym set they believed would fit that context . Figure 6 shows one sentence taken from the union of human annotator data. The complete set of annotated data is available in [citation removed].

Sentence: Moreover , Mr. Davis says , in some cases dad is always running interference , and that makes it _____ for his daughter to assume responsibility and learn from experience .
Human Suggestion: difficult
Human Suggestion: hard
Human Suggestion: tough
Human Suggestion: challenging
Computer Suggestion: difficult frequency: 522.0 ba:1 bb:1 bs:1 bis:1 - 100% certain
Computer Suggestion: hard frequency: 269.0 ba:1 bb:1 bs:1 bis:1 — 100% certain
Computer Suggestion: tough frequency: 5.0 ba:0 bb:0 bs:1 bis:1 — 50% certain
Computer Suggestion: trying frequency: 37.0 ba:0 bb:0 bs:0 bis:1 — 25% certain
Computer Suggestion: unmanageable frequency: 1.0 ba:0 bb:0 bs:0 bis:1 — 25% certain

Fig. 6. Example from the intersection of human annotated data. Certainty is based on feature agreement.

Table 3 summarizes our system's word selection compared to both the union and intersection of the human annotated data. The union combines the humans' annotations; the intersection is what they agreed upon. The second column provides the proportion of first suggestions by the system that is in the human annotators' (union/intersection) list; the third column gives the proportion of the humans' list that contains the system's suggestions (i.e., the intersection).

Table 3. Summary of system performance

Synonym Set	First suggestion correct	Correct word among suggestions
Difficult set (union)	92%	97%
Difficult set (intersection)	81%	94%
Little set (union)	86%	93%
Lttle set (intersection)	83%	92%

7 Contributions

By using copious amounts of very limited contextual linguistic data (only five words long) from the Google n-grams our system is able to achieve comparable results to prior research on previous testing data. Our system is believed to be the first of its kind able to make multiple word suggestions. Multiple word suggestions are believed to more accurately reflect human choice because in the majority of real-world linguistic contexts there is a high number of possibilities for near-synonym usage. This paper also presents new richer synonyms sets as well as a human annotated data with multiple word suggestions that is more conducive to testing this problem and should be of use in future. It contains a large number of words and is able to be run in real-time. When able to make multiple suggestions, it improved by 4.5% over previous human annotators who were also allowed multiple suggestions.

8 Future Work

Undoubtedly, through future work this method could be improved upon. There are several areas on which to focus future work. The n-gram dataset, part-of-speech tagging, particles, and the computer suggested word ranking system.

Working with the 2012 version of the Google n-gram database derived from books would most likely yield positive results. Our method caters toward speed and portability but if accuracy is the primary goal, we predict a substantial improvement by using a larger n-gram dataset.

The consideration of particles would allow the replacement of synonyms with multiword verb/particles and vice versa, e.g., *find* : *look up*. This inclusion would increase the functionality of our system greatly but may be made difficult because words can appear in between the verb and the particle, *look the word up*.

Part of speech tagging would address some of the problems that we observed in testing with the word *trying*, which was often used as a present participle complement of the verb *to be* rather than as an adjective which was desired.

In addition there are many of future applications our system could be used for and because of its small storage requirements and speed could be used on a variety of devices.

– To aid in translation such as in the Google translate application a major issue is finding the correct synonym to fit a certain context so the sentence still makes sense. *I got wet outside* rather than *I got rainy outside*.
– Automatic word usage correction on documents — to automatically scan a document and correct bad word usage based on a user set certainty (25% = one feature, 50% = two features, 75% = three features, 100%= all four features).
– Word usage correction in word processing — a real-time system running in the background of the word processor similar to spell and grammar check to notify the user of incorrect word usage and display a list of possible suggestions.

Acknowledgements

Appropriate acknowledgements will be provided.

References

1. Brants, T., Franz, A.: Web 1T n-grams version 1.1. Tech. rep., Google Research (2006)
2. Edmonds, P.: Choosing the word most typical in context using a lexical co-occurrence network. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics. pp. 507–509 (1997)
3. Google books n-grams. `http://storage.googleapis.com/books/ngrams/books/datasetsv2.html` (2012)
4. Inkpen, D.: A statistical model for near-synonym choice. ACM Trans. Speech Lang. Process. 4(1), 2:1–2:17 (2007)
5. Inkpen, D., Hirst, G.: Building and using a lexical knowledge-base of near-synonym differences. Computational Linguistics 32, 1–39 (2006)
6. Islam, A., Inkpen, D.: Near-synonym choice using a 5-gram language model. Research in Computing Sciences 46, 41–52 (2010)
7. Lam, Y.C.: Managing the google web 1T 5-gram with relational database. Journal of Education, Informatics, and Cybernetics pp. 1–6 (2010)
8. About Wordnet. `http://wordnet.princeton.edu` (2014)