Semantic Pleonasm Detection

Omid Kashefi **Andrew T. Lucas** **Rebecca Hwa**
School of Computing and Information
University of Pittsburgh
Pittsburgh, PA, 15260
{kashefi, andrew.lucas, hwa}@cs.pitt.edu

Abstract

Pleonasms are words that are redundant. To aid the development of systems that detect pleonasms in text, we introduce an annotated corpus of semantic pleonasms. We validate the integrity of the corpus with inter-annotator agreement analyses. We also compare it against alternative resources in terms of their effects on several automatic redundancy detection methods.

1 Introduction

Pleonasm is the use of extraneous words in an expression such that removing them would not significantly alter the meaning of the expression (Merriam-Webster, 1983; Quinn, 1993; Lehmann, 2005). Although pleonastic phrases may serve literary functions (e.g., to add emphasis) (Miller, 1951; Chernov, 1979), most modern writing style guides caution against them in favor of concise writing (Hart et al., 1905; Williams, 2003; Turabian, 2013; Gowers, 2014; Strunk, 1920).

An automatic pleonasm detector would be beneficial for natural language processing (NLP) applications that support student writing, such as grammar error correction (GEC) (Han et al., 2006; Rozovskaya and Roth, 2010; Tetreault et al., 2010; Dahlmeier and Ng, 2011), automatic essay grading (Larkey, 1998; Landauer, 2003; Ong et al., 2014), and intelligent writing tutors (Merrill et al., 1992; Aleven et al., 2009; Atkinson, 2016). Pleonastic phrases may also negatively impact NLP applications in general because they introduce an unnecessary complexity to the language. Their removal might facilitate NLP tasks such as parsing, summarization, and machine translation. However, automated pleonasm detection is a challenging problem, in parts because there is no appropriate resources to support the development of such systems. While some GEC corpora do annotate some words or phrases as "redundant" or "unnecessary," they are typically a manifestation of grammar errors (e.g., *we still have room to improve* **for** *our current welfare system*) rather than a stylistic redundancy (e.g., *we aim to* **better** *improve our welfare system*).

This paper presents a new Semantic Pleonasm Corpus (SPC), a collection of three thousand sentences. Each sentence features a pair of potentially semantically related words (chosen by a heuristic); human annotators determine whether either (or both) of the words is redundant. The corpus offers two improvements over current resources. First, the corpus filters for grammatical sentences so that the question of redundancy is separated from grammaticality. Second, the corpus is filtered for a balanced set of positive and negative examples (i.e., no redundancy). The negative examples may make useful benchmark data – because they all contain a pair of words that are deemed to be semantically related, a successful system cannot rely on simple heuristics, such as semantic distances, for discrimination. We evaluate the corpus in terms of inter-annotator agreement, and in terms of its usefulness for developing automatic pleonasm detectors.

2 Semantic Pleonasm

Although pleonasm is generally a semantic and rhetorical concept, it could have different aspects and be formed in different layers of language, including morphemic (e.g., "**irregardless**" (Berube, 1985)) and syntactic layers (e.g., "*the* **most unkindest** *cut of all*"). Detecting and correcting morphemic and syntactic pleonasms are more in the scope of GEC research, especially when they cause errors. Semantic pleonasm, on the other hand, is "a question of style or taste, not gram-

Proceedings of NAACL-HLT 2018, pages 225–230
New Orleans, Louisiana, June 1 - 6, 2018.

mar" (Evans and Evans, 1957). It occurs when the meaning of a word (or phrase) is already implied by other words in the sentence. For example, the following is a grammatical sentence that has a redundant word: *I received a* **free** *gift*. While writers might intentionally include the redundant word for emphasis, the overuse of pleonasm may weaken the expression, making it "boring rather than striking the hearer." (Fowler, 1994).

3 A Semantic Pleonasm Corpus

Semantic pleonasm is a complex linguistic phenomenon; to develop a useful corpus for it, we need to make some design decisions in terms of a trading off between the breadth and depth of our coverage.

3.1 Data Source

We want to start from a source that is likely to contain semantic redundancies. Because good writers are trained to guard against redundant phrasings, professionally written text from Project Gutenberg or the Wall Street Journal would not be appropriate. Because we want to separate the issues of grammaticality from redundancy, learner corpora would also not be appropriate. A data source that seems promising is amateur product reviews. The writers tend to produce more emotional prose that are at times exasperated or gushing; the writing is more off-the-cuff and casual, and may contain more redundancy. Ultimately, we chose to work with restaurant reviews from Round Seven of the Yelp Dataset Challenge[1] because it is widely distributed.

3.2 Filtering

Although redundant words and phrases occur frequently enough that exhortations to excise them is a constant refrain in writing guides, most sentences still skew toward not containing pleonasms. Annotating all sentences would dilute the impact of the positive examples, further complicate the annotation scheme, and increase the cost of the corpus creation. Thus, we opt to construct a balanced corpus of positive and negative examples for a specific kind of redundancy in a specific configuration. In particular, we extract all sentences that contained a pair of adjacent words that are likely to be semantically similar. We restrict our attention to adjacent word pairs to increase the chance of finding redundancy, since semantically related words that are farther apart are more likely to have different syntactic and semantic roles. To determine semantic similarity, we use the TextBlob Python interface[2], which, for a given word, provides access to WordNet synsets (Miller, 1995) corresponding to each of the word's senses. We compare each pair of adjacent words in the dataset to see whether they share any synsets. Since WordNet serves as a coarse filter, we need to further improve recall. We select any sentences that contains a pair of adjacent words such that one of the words has a synset that is *similar to* a synset of the other word. TextBlob provides this "similar to" functionality, which finds synsets that are close to a given synset in WordNet's taxonomy tree. (note, however, that these words may not be used in those senses in the sentence). Applying these filtering rules, we are able to eliminate a large percentage of sentences that do not contain semantic redundancy; the method also help us identify a pair of words in each sentence that is likely to have a redundancy. In the second step of filtering, we manually removed sentences that contained obvious grammatical mistakes.

3.3 Annotation

We set up a Amazon Mechanical Turk service to determine whether the potentially redundant word pairs are actually redundant. Because we want to build a balanced corpus, we first perform a quick internal first pass, marking each sentence as either "possibly containing redundancy" or "probably not containing redundancy" so that we can distribute the instances to the Turkers with equal probability (they do not see our internal annotations). The Turkers are given six sentences at a time, each containing a highlighted pair of words. The workers have to decide whether to delete the first word, the second word, both, or neither. Then, they indicate their confidence: "Certain," "Somewhat certain," or "Uncertain." Lastly, they are given the opportunity to provide additional explanations. Each sentence has been reviewed by three different workers. For about ninety percent of the sentences, three annotations proved sufficient to achieve a consensus. We collect a fourth annotation for the remaining sentences, and are then able to declare a consensus.

[1] https://www.yelp.com/dataset/challenge

[2] http://textblob.readthedocs.io/en/dev/

Consensus Level	Fleiss's Kappa
Word Level	0.384
Sentence Level	0.482

Table 1: Inter-Annotator Agreement

One		Both	Neither	Total
First	Second			
955	765	16	1,283	3,019
32%	25%	1%	42%	100%
57%				

Table 2: Statistics of the Semantic Pleonasm Corpus

A Few Examples

- Sentence: *Freshly squeezed and no additives, just* **plain pure** *fruit pulp.*
  Consensus: plain is redundant.
- Sentence: *It is clear that I will never have another* **prime first** *experience like the one I had at Chompies.*
  Consensus: neither word is redundant.
- Sentence: *The dressing is absolutely* **incredibly fabulously** *flavorful!*
  Consensus: both words are redundant.

3.4 Inter-Annotator Agreement

Because our corpus is annotated by many Turkers, with some labeling only a handful of sentences while others contributed hundreds, the typical pair-wise inter-annotated agreement is not appropriate. Instead, we compute Fleiss's Kappa (Fleiss, 1971), which measures the degree of agreement in classification over what would be expected by chance for more than two annotator.

We analyze agreements at two levels of granularity: *word level* indicates the consensus on whether the first, second, both, or neither of the candidates is pleonastic; *sentence level* indicates the consensus on whether a sentence has a pleonastic construction.

Table 1 shows that annotators are more likely to agree whether a sentence contains a pleonasm than exactly which words should be considered redundant. In many cases, a majority consensus is achieved with one annotator disagreeing with the others. The result suggests that when there is a single word redundancy, removing either of the synonyms could be appropriate.

3.5 Properties

The final dataset consists of 3,019 sentences. Their final labels are based on a majority consensus: 1,283 sentences are marked as not having a redundant word; 1,720 sentences are marked as containing a single word redundancy; and for 16 sentences, both words are marked as redundant. Table 2. shows the statistics of annotators consensus. The corpus, including all annotations and the final consensus, is available in JSON format from `http://pleonasm.cs.pitt.edu`

4 Automatic Pleonasm Detection

Given our design choices, the current SPC is not a large corpus; we posit that it can nonetheless serve as a valuable resource for developing systems to detect semantic pleonasm. For example, the earlier work of Xue and Hwa (2014) might have benefited from this resource. They wanted to detect the word in a sentence that contributes the least to the meaning of the sentence; however, their experiments were hampered by a mismatch between their intended domain and the corpus they evaluated on – while their model estimated a word's semantic redundancy, their experiments were performed on NUCLE (Dahlmeier et al., 2013), a learner corpus that focused more on grammatical errors. Moreover, since their detector always returned the word with the lowest meaning contribution score, they only evaluated their model on sentences known to contain an unnecessary word; without appropriate negative examples, it is not clear how to apply their system to sentences with no redundancy. These are two use-case scenarios that the SPC may address. To verify our claim, we will first compare the performances of several word redundancy metrics, including a replication of the metric of Xue and Hwa, on our corpus with their performances on NUCLE. We will then show that the SPC can train a classifier that predicts whether a sentence contains semantic pleonasm.

4.1 Pleonastic Word Detection

This experiment focuses on the positive examples – the methods under evaluation are all metrics for detecting the most redundant word from sentences known to contain one. We compare the performances of different word detectors under SPC and

NUCLE. Note that our experimental goal is not to obtain a method that reports a high accuracy on SPC (we do not want a corpus that overfits to some particular method). Rather, it is to demonstrate that the human-annotated SPC captures aspects of semantic redundancy that are not available in other resources.

In order to shed lights on the differences between SPC and NUCLE, we compare them using detectors that are formulated from different strategies. First, we have replicated the metric proposed by Xue and Hwa, which consists of two main components: a language model and a word meaning contribution model that is derived from word alignments from machine translation[3]. This method is the most focused on lexical semantic, so we expect it to be better at detecting redundant words on the SPC. Next, we have implemented three simple metrics: SIM computes the semantic similarity between a full sentence and that sentence with the target word removed[4]; GEN estimates the degree to which a word is general (therefore more likely to be redundant) by its number of synonyms; and SMP estimates the simplicity of a word based on an implementation of the Flesch-Kincaid readability score (Kincaid et al., 1975). Of these, only SIM directly models semantics; we expect it to be better at detecting redundant words on the SPC than the two other, more general, metrics. Finally, as a point of contrast, we consider a GEC system using *languagetools*[5] (Naber, 2003); we expect the GEC system to be better at detecting grammar error related redundancy found on NUCLE than cases of semantic redundancy found in the SPC.

To conduct the experiment, we selected 1,140 NUCLE sentences that contain one local redundancy ($RLOC$) error; for SPC, 1,720 sentences with one semantic pleonasm are used. Table 3 shows the accuracy of each method under both corpora. Our implementation of Xue and Hwa's model replicates their reported outcome with NUCLE, and, as expected, their method is more successful on the SPC. All three simple metrics are more successful at picking out redundant word on the SPC than NUCLE, with SIM showing a bigger difference than the other two. Comparing the four methods' between corpora differences, we see that the method of Xue&Hwa has the most to gain, perhaps because it has the strongest domain mismatch. Yet, a combination of all four metrics results in an improved accuracy of 39.4%, suggesting that the four strategies capture different aspect of semantic redundancy. That this highest achieving accuracy is still quite low suggests that there is ample room for improvement in terms of word detector development. In contrast, the GEC method performed much better on NUCLE (11.9%) than on the SPC (4.7%). Taken as a whole, these results suggest that the SPC, while small, is a better fit for the task of detecting semantic redundancy than NUCLE.

Method	NUCLE	SPC
Xue&Hwa	22.8%	31.7%
SIM	11.1%	16.6%
GEN	9.6%	13.3%
SMP	16.1%	20.6%
SIM + SMP + GEN	18.2%	27.6%
ALL	31.1%	39.4%
GEC	11.9%	4.7%

Table 3: The accuracy of detecting the redundant word in sentences with different methods under two corpora: NUCLE and SPC. ALL is a composite metric from the other four: $Xue\&Hwa + SIM + SMP + GEN$.

4.2 Sentential Pleonasm Detection

All the methods shown in the previous experiment are metrics that assign a redundancy score to each word within a sentence; they still have to be incorporated into an outer classifier to determine whether the sentence indeed contains a pleonasm. A corpus of naturally occurring text is unsuitable for training the classifier because the distribution is heavily skewed toward the *no redundancy* case. Random down-sampling is also not ideal because some might be too obvious (e.g., very short sentences). SPC addresses this problem by filtering for challenging negative cases: sentences that contain a pair of words that are heuristically deemed to be semantically related, but are not judged to

[3] In our re-implementation, the language model is trained on a portion of English Gigaword (Graff et al., 2003) using KenLM (Heafield, 2011); the word alignments are derived from Bing's English-French Translator

[4] using *sense2vec* word-embeddings (Trask et al., 2015)

[5] `https://languagetool.org/`

Feature	Description
UG	the one-hot representation (Harris and Harris, 2010) of unigrams of the sentence
TG	the one-hot representation of trigrams of the sentence
TFIDF	the one-hot representation of smoothed TFIDF tuples of trigrams of the sentence
WSTAT	[max(ALL), avg(ALL), min(ALL), Len(s), LM(s)]

Table 4: Features for sentential level pleonasm detection. ALL represents the collection of word-level metrics: $Xue\&Hwa$, SIM, GEN, and SMP; Len(s) is the number of words in sentence s; LM(s) is the trigram probability for sentence s.

Baseline	SPC	
	MaxEnt	Naive Bayes
UG	79.2	88.4
TG	79.9	88.8
TFIDF	83.0	90.5
WSTAT	63.1	53.2
WSTAT+UG	82.3	89.2
WSTAT+TG	83.7	89.3
WSTAT+TFIDF	84.5	92.2

Table 5: The accuracy of a binary classifier using different feature set to predict whether a sentence contains a pleonastic construction.

be redundant by human annotators. In this experiment, we use SPC to train a binary classifier; our feature set is summarized in Table 4.

To train the classifiers, we performed 5-fold cross-validation on the full SPC corpus. We experimented with both a Maximum Entropy and a Binomial Naive Bayes binary classifier. We considered the number of features from χ^2 test, regularization coefficient, the choice of penalty function and solver as hyperparameters and optimized them using the Particle Swarm algorithm (Clerc and Kennedy, 2002) in the Optunity[6] optimizer package.

Table 5 presents the results. We observe that the three features that directly encode the words of the sentence are more relevant (UG, TG, $TFIDF$) than the group of statistics over the word redundancy metrics ($WSTAT$). For our corpus size, Naive Bayes seems to converge faster to the minimum error rate than MaxEnt (Ng and Jordan, 2002). In combination, $WSTAT + TFIDF$ gave the highest accuracy, at around 92%. This result also reinforces our inter-annotator agreement rate, suggesting that determining whether a sentence contains a semantic pleonasm is easier than deciding which word is pleonastic.

[6] https://github.com/claesenm/optunity

5 Conclusion

We have introduced a semantic pleonasm corpus in which each sentence contains a word pair that is potentially semantically related. These sentences are reviewed by human annotators, who determine whether any of the words are redundant. Our corpus offers two main contributions. First, as a corpus that focuses on semantic similarity, it provides a more appropriate resource for systems that aim to detect stylistic redundancy rather than grammatical errors. Second, as a balanced corpus of positive and near-miss negative examples, it allows systems to evaluate their ability to detect "no redundancy."

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This material is based upon work supported by the National Science Foundation under Grant Number #1735752.

References

Vincent Aleven, Bruce M Mclaren, Jonathan Sewall, and Kenneth R Koedinger. 2009. Example-Tracing Tutors: A New Paradigm for Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education* 19(2):105–154.

Robert Kenneth Atkinson. 2016. *Intelligent tutoring systems: Structure, applications and challenges*. Nova Science Publishers, Inc.

Margery Berube. 1985. *The American Heritage Dictionary: Second College Edition*. Houghton Mifflin.

Ghelly V Chernov. 1979. Semantic Aspects of Psycholinguistic Research in Simultaneous Interpretation. *Language and Speech* 22(3):277–295.

Maurice Clerc and James Kennedy. 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1):58–73.

Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical Error Correction with Alternating Structure Optimization. In *Proceedings of the 49th Annual Meeting of the ACL-HLT*. pages 915–923.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. pages 22–31.

Bergen Evans and Cor Nelia Evans. 1957. *A Dictionary of Contemporary American Usage*. Random House, Inc.

Joseph L Fleiss. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin* 76(5):378.

Henry Watson Fowler. 1994. *A Dictionary of Modern English Usage*. Wordsworth Editions.

Ernest Gowers. 2014. *Plain Words*. Penguin UK.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English Gigaword. *Linguistic Data Consortium, Philadelphia* 4:1.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering* 12(02):115–129.

David Harris and Sarah Harris. 2010. *Digital Design and Computer Architecture*. Morgan Kaufmann.

Horace Hart, James Augustus Henry Murray, and Henry Bradley. 1905. *Rules for Compositors and Readers at the University Press, Oxford*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. pages 187–197.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel.

Thomas K Landauer. 2003. Automatic Essay Assessment. *Assessment in Education: Principles, Policy & Practice* 10(3):295–308.

Leah S Larkey. 1998. Automatic Essay Grading Using Text Categorization Techniques. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pages 90–95.

Christian Lehmann. 2005. Pleonasm and Hypercharacterisation. *Yearbook of Morphology 2005* pages 119–154.

Inc Merriam-Webster. 1983. *Webster's Ninth New Collegiate Dictionary*. Merriam-Webster.

Douglas C Merrill, Brian J Reiser, Michael Ranney, and J Gregory Trafton. 1992. Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems. *The Journal of the Learning Sciences* 2(3):277–305.

George A Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38(11):39–41.

George Armitage Miller. 1951. *Language and Communication*. McGraw-Hill.

Daniel Naber. 2003. *A Rule-Based Style and Grammar Checker*. B.S. Thesis, Bielefeld University.

Andrew Y Ng and Michael I Jordan. 2002. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes. In *Advances in Neural Information Processing Systems*. pages 841–848.

Nathan Ong, Diane Litman, and Alexandra Brusilovsky. 2014. Ontology-Based Argument Mining and Automatic Essay Scoring. In *Proceedings of the First Workshop on Argumentation Mining*. pages 24–28.

Arthur Quinn. 1993. *Figures of Speech: 60 Ways to Turn a Phrase*. Psychology Press.

Alla Rozovskaya and Dan Roth. 2010. Generating Confusion Sets for Context-Sensitive Error Correction. In *Proceedings of the EMNLP*. pages 961–970.

William Strunk. 1920. *The Elements of Style*. New York: Harcourt, Brace and Howe.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using Parse Features for Preposition Selection and Error Detection. In *Proceedings of the ACL Short Papers*. pages 353–358.

Andrew Trask, Phil Michalak, and John Liu. 2015. sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings. *arXiv preprint arXiv:1511.06388* .

Kate L Turabian. 2013. *A Manual for Writers of Research Papers, Theses, and Dissertations: Chicago Style for Students and Researchers*. University of Chicago Press.

Joseph M Williams. 2003. *Style*. Longman.

Huichao Xue and Rebecca Hwa. 2014. Redundancy Detection in ESL Writings. In *EACL*. pages 683–691.