Homonym Detection For Humor Recognition In Short Text

Sven van den Beukel
Faculteit der Bèta-wetenschappen
VU Amsterdam, The Netherlands
sbl530@student.vu.nl

Lora Aroyo
Faculteit der Bèta-wetenschappen
VU Amsterdam, The Netherlands
l.m.aroyo@vu.nl

Abstract

In this paper, automatic homophone- and homograph detection are suggested as new useful features for humor recognition systems. The system combines style-features from previous studies on humor recognition in short text with ambiguity-based features. The performance of two potentially useful homograph detection methods is evaluated using crowdsourced annotations as ground truth. Adding homophones and homographs as features to the classifier results in a small but significant improvement over the style-features alone. For the task of humor recognition, recall appears to be a more important quality measure than precision. Although the system was designed for humor recognition in oneliners, it also performs well at the classification of longer humorous texts.

1 Introduction

Humor has the potential to help form, strengthen and maintain human relationships and could thus bring humans and computers closer to each other. It helps regulate conversations, builds trust between partners, facilitates self-disclosure and it is an important factor in social attraction (Nijholt et al., 2003). Furthermore, humans react in the same way to computers as they do to other human beings when it comes to psycho-social phenomena (Morkes et al., 1998; Reeves and Nass, 1996). Experiments have shown that people that received a joke, perceived the computer they interacted with as more likable and competent, reported greater cooperation and responded more sociable (Morkes et al., 1998). Automatic humor recognition could help computers respond more appropriately, making human-computer interaction feel more natural and enjoyable.

This paper focuses on humor recognition of written oneliners, which in this study are defined as short jokes that are at most 140 characters long. The popularity of Twitter has likely caused an increase in availability of both humorous and non-humorous texts shorter than 140 tokens. The choice for oneliners increases difficulty of humor recognition as they contain less contextual information than longer humorous texts. The built classifier is also tested on humor recognition in larger texts. In this study, features that capture text style are selected from the State-of-the-Art on humor recognition in oneliners (Mihalcea and Strapparava, 2005), cartoon captions (Radev et al., 2015) and tweets (Zhang and Liu, 2014) and are combined with newly suggested ambiguity features. When referring to "The State-of-the-Art", we refer to Mihalcea and Strapparava (2005). This allows us to evaluate the usefulness of the style-features for application on humor recognition in oneliners (rather than cartoons or tweets), as well as the potential of automatic homophone and homograph detection as signalers of ambiguity, and subsequently humor.

The release of the datasets and code that were used (Appendix A) are also a valuable contribution, since it allows others to replicate the experiments and to explore further directions. The humorous oneliners and Reuters datasets themselves are not publicly released to prevent potential copyright infringements, but these can be requested from the authors. Two methods for detecting homophones and homographs are designed to detect ambiguity, after which the performance of the proposed methods is evaluated. In the remainder of this document these features might be referred to as "homonyms", the category of words to which homographs and homophones belong. The deployment of content-based features (e.g. LSA) are outside the scope of this study, despite their previously reported usefulness (Mihalcea and Strapparava, 2005; Sjöbergh and Araki, 2007). The per-

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 286–291
Brussels, Belgium, October 31, 2018.
https://doi.org/10.18653/v1/P17

formance achieved through content-based features might be unsustainable over time due to the changing nature of language. Style- and ambiguity-features have the potential to make classification results more sustainable. At the end of this paper, four research questions are answered.

1. How should high quality data for training a humor recognition system be gathered?
2. Which automatic homograph recognition method adds the highest information gain for humor recognition in oneliners?
3. Does the presence of automatically extracted homophones and homographs improve the accuracy of humor recognition in oneliners?
4. Can the proposed classification framework be used for recognizing humor in longer texts?

2 Related work

First of all, in this study the incongruity-resolution theory of humor is used as a frame for selecting useful stylistic features. It is argued to be the most influential theory used to study humor and laughter (Mulder and Nijholt, 2002). When one examines jokes according to the incongruity frame, two concepts within the joke are examined through one frame. When the recipient of the joke notices that the frame actually only applies to one of the objects, the difference between the two objects and the frame becomes apparent (incongruity). The humorous situation occurs when the recipient recognizes the congruous resolution of the apparent incongruity. This theory fits this study best, since it explains the structure of a joke. First there is an incongruity, then a congruous resolution is provided (Gruner, 2000).

2.1 Stylistic features

The stylistic features used in the State-of-the-Art are alliteration, antonymy and adult slang (Mihalcea and Strapparava, 2005). In this study, the features capturing alliteration and rhyme are separated, which was found to be useful by Zhang and Liu (2014). The reason these stylistic features are informative, could be that oneliners use rhyme or alliteration to create expectation and - if humorous - to break it. The expectation creates incongruity, which is resolved through breaking it. Secondly, negations (Mihalcea and Pulman, 2007) and antonyms (Mihalcea and Strapparava, 2005) signal incongruity by having contradictions within a sentence. Thirdly, humorous oneliners were found to contain adult slang. Whereas the State-of-the-Art represented adult slang using sex-related words, insults and vulgar words are included in this study as well. Moreover, researchers have reported that negative and positive sentiment can help distinguish humorous from less humorous samples (Mihalcea and Pulman, 2007; Radev et al., 2015). Furthermore, humorous texts generally have higher sentiment polarity than non-humorous texts, which was found useful for classifying humorous tweets (Zhang and Liu, 2014). Additionally, the latter study found that the ratios of several Part of Speech tags are informative.

2.2 Ambiguity detection

Some types of humor (e.g. wordplay), owe their funniness directly to the presence of ambiguity (Taylor and Mazlack, 2004). In order to identify wordplays, the computer has to combine general knowledge of the world and of pronunciation. Wordplays surprise the recipient of the joke by breaking an expectation. This can be achieved through homographs (e.g. *"Cliford: The Postmaster General will be making the toast. Woody: Wow, imagine a person like that helping out in the kitchen!" (Taylor and Mazlack, 2004)*, in which toast is written the same yet has multiple meanings). Another possibility is the use of homophones, which are words that sound alike yet have different meanings (e.g. *"What is everybody's favorite aspect of mathematics? Knot theory, that's for sure."*, in which "knot" and "not" sound alike). Homophones are not necessarily spelled the same. Example previous attempts at ambiguity detection include a count of the number of senses available for a word (Barbieri and Saggion, 2014; Sjöbergh and Araki, 2007) and the number of parses possible for a sentence (Sjöbergh and Araki, 2007). Since ambiguity is such a complex problem to solve, there is room for improvement. Kao et al. (2015) have recently shown that homophones can be humorous, but only if both interpretations of the homophone are supported by the other words in the sentence. The more distinct the support for the multiple interpretations, the bigger the incongruity-resolution and thus the more humorous the oneliner is perceived. A similar observation has been reported for homographs (McHugh and Buchanan, 2016). However, to our knowl-

edge no automatic homophone- or homograph-detection methods exist yet.

3 Approach

All sentences are at most 140 characters long, to prevent classification based on sentence length. In this study we used one humorous dataset, two non-humorous datasets that are stylistically similar to it (Reuters news headlines and English proverbs), and a third non-humorous dataset that has content comparable to the humorous dataset (wikipedia sentences).

3.1 Data gathering

Reuters news headlines were selected as they share the properties with humorous oneliners of being concise sentences that attract the attention of the reader to transfer a message. The second stylistically similar, non-humorous dataset consists of English proverbs. Proverbs are short texts that transmit facts or experiences of everyday life that many people consider to be true. Finally, the negative set containing short wikipedia sentences attempts to represent real-world scenarios. This set replaces the British National Corpus or the Open Mind Common Sense corpus used in the State-of-the-Art, which we were unable to collect.

Humorous oneliners are collected with a web-scraper designed for five manually selected websites dedicated to jokes[1]. The resulting dataset contains 12,046 oneliners and 5,606 jokes longer than 140 characters.

News Headlines are scraped from the website of publishing agency Reuters and were retrieved on August 15th, 2017. Headlines from multiple categories ("Business", "Politics", "World" and "Technology") were extracted to prevent topic-based classification. The full dataset contains 13,798 headlines.

English proverbs were collected manually [2], and due to scarcity this set is limited to 1,019 samples. The classifiers trained with proverbs as non-humorous samples, use an equal amount of humorous samples to prevent overfitting.

Wikipedia sentences were retrieved from a dataset provided in a study on text simplification (Kauchak, 2013), of which 12,046 items are selected based on size and content similarity (TF-IDF). This dataset is expected to be the hardest to classify due to the similarity in content with the humorous oneliners.

3.2 Detecting style and ambiguity

This paragraph lists the approaches for extracting the style- and ambiguity features. Since the approaches for extracting homonyms are designed from the ground up, they require evaluation.

Alliteration & Rhyme presence is measured through the CMUDict[3] phoneme dictionary. For alliterations, n-grams are considered an alliteration chain only if the first phoneme of a word is the same as the first one of one of the two next words. Rhymes are identified the same way, but consider the last phonemes rather than the first ones. For example, *goal* and *Glasgow* alliterate, and *score* rhymes with *more*. For both alliteration and rhyme, one feature is created containing the number of chains in a sentence, and a second consisting of the length of the longest chain in the sentence, divided by the number of words.

Sentiment polarity is the total sentiment score of a sentence, calculated using the Senticnet 4 package for Python (Cambria et al., 2016). The sentiment intensity scores ranging from very negative (-1) to very positive (+1) are used to calculate the total sentiment polarity of a sentence. A sentence that has both positive and negative parts in it, might result in a neutral score. In order to account for this, a second feature is introduced using only natural numbers. For example, a oneliner scoring -2 and +2 sentiment scores, is represented in the second feature with a value of 4.

Part of Speech-tag ratios are calculated using Stanford CoreNLP to tag sentences with Treebank pos-tags (Manning et al., 2014) and dividing the number of occurrences for each POS-category by the number of words in a sentence. The POS-tag categories included are pronouns, verbs, common nouns, proper nouns and modifiers.

Antonymy presence is evaluated using the WordNet "Antonymy"-relationship. Since not all antonyms are listed (Mihalcea and Strapparava, 2005), this set is expanded by also checking whether the antonyms of synonyms of any adjectives are present.

Adult Slang is identified in text, by putting all synsets that are hyponyms of the WordNet synsets 'sexuality' and 'sexual_activity' up to a depth of

[1] funnyshortjokes.com, goodriddlesnow.com, laughfactory.com, onelinefun.com and unijokes.com

[2] www.english-for-students.com and www.citehr.com

[3] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

three layers of hyponyms in a lexicon, and comparing the words in the sentence to it. Moreover, the definitions of each remaining word are scanned for phrases that signal adult slang, such as 'offensive word', 'obscene word' and 'vulgar term'.

Negations are identified by checking whether the word "not" or contraction "..n't" occurs.

Homophones are recognized using CMUDict to find words that have similar pronunciations. For each word in a sentence, another word is sought with the same pronunciation. A small experiment showed that this approach detects over 83% of the homephones found on an expert-created list[4], while capturing more than are on the list.

Homographs are identified using two methods. The first method matches words from sentences to a list of 160 common homographs retrieved from Wikipedia. The second method uses WordNet to extract the definitions of all senses found for a word and only keeps those definitions with no overlap in used vocabulary. A word is considered a homograph if more than two definitions remain.

3.3 Crowdsourcing homograph annotation

The performance of the two proposed homograph detection algorithms is measured by comparing the accuracy on a dataset containing 301 sentences with annotated homographs. The users of crowdsourcing platform Crowdflower[5] were presented with a sentence, and a list of answer options on clickable buttons. The 301 annotated sentences were randomly selected from the oneliners, reuters and wiki datasets and excluded for training.

For assessment of annotation quality, three metrics from the CrowdTruth approach were used (Dumitrache et al., 2015). This approach helps to extract more information from annotations by taking both annotator agreement and -disagreement into account, requiring less annotations for high quality results. The formulas for the used metrics can be found on GitHub[6].

First of all, the Media Unit Quality score (UQS) captures the level of agreement in annotation of a media unit. This metric helps identify ambiguity in the task of annotating specific sentences. Sentences that are hard to annotate, have a low UQS. In this particular annotation task, this means that sentences with a low UQS likely contain homographs that are difficult to recognize or that are debatable. Secondly, the Worker Quality Score (WQS) assigns a score to each worker based on its annotation agreement with others that worked on the same sentences (Worker-Worker Agreement) and a workers' disagreement compared to the crowd, on a sentence basis. By using the weighted average, poor annotations of sentences that were found to be difficult to classify, have a lower impact on the final WQS of a worker. Finally, the UQS and WQS are combined into a weighted annotation score (Unit Annotation Score), giving better annotators more influence on the final annotation score of a sentence. The results are reported in sections 4.1 and 4.2.

3.4 Machine Learning algorithms

Three machine learning algorithms are deployed in this study, consisting of one Naive Bayes (Bernoulli NB) implementation and two Support Vector Machines (SVMs) with a linear and RBF kernel respectively. The main advantage NB classifiers have over their more sophisticated counterparts are its speed and reduced complexity. On the other hand, SVMs (Burges et al., 1996) outperformed other commonly used algorithms such as Naive Bayes, K-NN and C4.5 Decision Tree learners at the widely used benchmark task of text categorization of Reuters data (Joachims, 1998).

3.5 Experimental Setting

In the first experiment, all the style-features are used for training the classifiers. The classifier performance is reported by its average accuracy over 30 runs using 10-fold cross-validation, to minimize variability in results. This is repeated once with homographs extracted using the list-approach and once with the WordNet approach. Comparison with the State-of-the-Art is not useful, since different datasets were used.

4 Results

4.1 Homograph annotation

A total of 221 out of 301 sentences have a UQS below 0.5, meaning they were difficult to annotate. Since only people from natively English-speaking countries were invited, homograph annotation seems to be a difficult task for humans. The WQS are also low, with the best worker reaching

[4] www.singularis.ltd.uk/bifroest/misc/homophones-list.html

[5] www.elite.crowdflower.com

[6] https://github.com/CrowdTruth/CrowdTruth-core/blob/master/tutorial/ CrowdTruth%20metrics%202.0%20documentation.ipynb

a score of 0.7 and 70 workers achieving a score lower than 0.3. The annotators achieving a WQS lower than 0.1 are most likely spam-workers. For the Media Unit Annotation Score, we find 248 words with a score higher than 0.5 that are thus labeled a homograph.

4.2 Homograph recognition performance

The performance results of the two homograph recognition methods is reported in Table 1. The acceptance threshold of 0.5 indicates that only words with a weighted annotation value higher than 0.5 are labeled as homographs (weighted majority vote). The fixed list of homographs performs rather well on precision and accuracy, as the data contains much more non-homographs than it does homographs. The poor recall however, suggests that the list contains an insufficient number of homographs. Although its precision and accuracy are lower, the WordNet approach results in a higher recall and f-measure, but suffers from a low precision due to its high number of false positives.

Table 1: Homograph recognition results

	Homograph list	WordNet
Precision	82.6	35.3
Recall	8.2	82.5
F-Measure	14.9	49.5
Accuracy	85.9	74.6

4.3 Experiments

The results for the experiments are reported in Table 2. The table shows, per column and in this order, the results using 14) only style-features, 15) features in 14 + homophones, 16L) features in 15 + list-matched homographs and finally 16W) features in 15 + the WordNet-homograhps. Bold results have a significantly higher mean accuracy when compared with featureset 14 with probability $P \leq 0.025$. The results of the system trained on oneliners an short wikipedia sentences and tested on humorous- and Wikipedia-texts longer than 140 characters, achieved a mean accuracy of 87.14%. All the results reported in Table 2 were achieved using the overall best performing classification algorithm (Linear SVM).

5 Discussion

The first research question concerned how high-quality data for training a humor recognition system should be gathered. Designing webscrapers targeting dedicated websites resulted in a dataset containing much less noise than the seedlist-webscraping approach reported in the State-of-the-Art (+-2% vs. +- 9% in a random 200 sentence sample) (Mihalcea and Strapparava, 2005). The second goal was to identify the best automatic homograph recognition method. For the task of humor recognition, the WordNet approach significantly outperforms the fixed list approach, which could suggest that recall is more important than precision for this task. The third goal was to evaluate whether automatically extracted homophones and homographs improve the accuracy of humor recognition in oneliners. Significant improvement in classification accuracy was found for homographs extracted through the WordNet approach, but not for homophones. Finally, the classifier trained on humorous and non-humorous oneliners performed well on humor classification in texts longer than 140 tokens (87.14% accuracy), suggesting the features are robust to variations in sentence length.

Table 2: Mean accuracy for each experiment

Featureset	14	15	16L	16W
Reuters	91.16%	91.11%	91.10%	**91.45%**
Wikipedia	69.66%	69.74%	69.66%	**69.94%**
Proverbs	75.78%	**75.98%**	**75.97%**	**76.91%**

In future work, it might be interesting to find out through feature selection which features are most informative. Although the homophone detection seems to work well, homophone presence in a sentence does not seem to hold significant predictive value without a measure of strength of support for different senses of the homophone in question.

6 Conclusions

This paper presents a method (and code, see Appendix A) for gathering high-quality training data, a homograph recognition evaluation set and a set of features that can be used alongside content-features to achieve a robust high classification performance. Homographs help detect ambiguity in sentences, which in turn was found to slightly increase classification performance. Homophone detection is possible, but does not yet add significant predictive value in its current implementation. A humor recognition classifier trained on oneliners can also accurately label longer texts.

References

Francesco Barbieri and Horacio Saggion. 2014. Automatic detection of irony and humour in twitter. In *ICCC*, pages 155–162.

Christopher JC Burges et al. 1996. Simplified support vector decision rules. In *ICML*, volume 96, pages 71–77. Citeseer.

Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Bjoern Schuller. 2016. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2666–2677. The COLING 2016 Organizing Committee.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2015. Achieving expert-level annotation quality with crowdtruth. In *Proc. of BDM2I Workshop, ISWC*.

Charles R Gruner. 2000. *The game of humor: A comprehensive theory of why we laugh*. Transaction publishers.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Justine T Kao, Roger Levy, and Noah D Goodman. 2015. A computational model of linguistic humor in puns. *Cognitive science*.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Tara McHugh and Lori Buchanan. 2016. Pun processing from a psycholinguistic perspective: Introducing the model of psycholinguistic hemispheric incongruity laughter (m. phil). *Laterality: Asymmetries of Body, Brain and Cognition*, 21(4-6):455–483.

Rada Mihalcea and Stephen Pulman. 2007. Characterizing humour: An exploration of features in humorous texts. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 337–347. Springer.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

John Morkes, Hadyn K Kernal, and Clifford Nass. 1998. Humor in task-oriented computer-mediated communication and human-computer interaction. In *CHI 98 Cconference Summary on Human Factors in Computing Systems*, pages 215–216. ACM.

Matthijs P Mulder and Antinus Nijholt. 2002. Humour research: State of art.

Anton Nijholt, Oliviero Stock, Alan Dix, and John Morkes. 2003. Humor modeling in the interface. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pages 1050–1051. ACM.

Dragomir Radev, Amanda Stent, Joel Tetreault, Aasish Pappu, Aikaterini Iliakopoulou, Agustin Chanfreau, Paloma de Juan, Jordi Vallmitjana, Alejandro Jaimes, Rahul Jha, et al. 2015. Humor in collective discourse: Unsupervised funniness detection in the new yorker cartoon caption contest. *arXiv preprint arXiv:1506.08126*.

Byron Reeves and Clifford Nass. 1996. *How people treat computers, television, and new media like real people and places*. CSLI Publications and Cambridge university press.

Jonas Sjöbergh and Kenji Araki. 2007. Recognizing humor without recognizing meaning. In *Proceedings of the 7th International Workshop on Fuzzy Logic and Applications: Applications of Fuzzy Sets Theory*, WILF '07, pages 469–476, Berlin, Heidelberg. Springer-Verlag.

Julia M Taylor and Lawrence J Mazlack. 2004. Computationally recognizing wordplay in jokes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.

Renxian Zhang and Naishi Liu. 2014. Recognizing humor on twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 889–898, New York, NY, USA. ACM.

Appendix A. Github project depository

The code and datasets are available here: https://github.com/svenvdbeukel/Short-text-corpus-with-focus-on-humor-detection

Appendix B. Link to supplementary information

Supplementary information useful for reproduction of the described experiments can be found by copying the following link: http://bit.ly/2MuVQg1Humor