SRHR at SemEval-2017 Task 6: Word Associations for Humour Recognition

Andrew Cattle **Xiaojuan Ma**
Hong Kong University of Science and Technology
Department of Computer Science and Engineering
Clear Water Bay, Hong Kong
{acattle,mxj}@cse.ust.hk

Abstract

This paper explores the role of semantic relatedness features, such as word associations, in humour recognition. Specifically, we examine the task of inferring pairwise humour judgments in Twitter hashtag wars. We examine a variety of word association features derived from the University of Southern Florida Free Association Norms (USF) (Nelson et al., 2004) and the Edinburgh Associative Thesaurus (EAT) (Kiss et al., 1973) and find that word association-based features outperform Word2Vec similarity, a popular semantic relatedness measure. Our system achieves an accuracy of 56.42% using a combination of unigram perplexity, bigram perplexity, $\text{EAT}^{\text{tweet-avg}}_{\text{difference}}$, $\text{USF}^{\text{max}}_{\text{forward}}$, $\text{EAT}^{\text{word-avg}}_{\text{difference}}$, $\text{USF}^{\text{word-avg}}_{\text{difference}}$, $\text{EAT}^{\text{min}}_{\text{forward}}$, $\text{USF}^{\text{tweet-max}}_{\text{difference}}$, and $\text{EAT}^{\text{min}}_{\text{backward}}$.

1 Introduction

What makes something funny? Humour is very personal; what is funny to one person may not be funny to another. Yet, there are still certain works which seem to have widespread appeal, from comedian Louis C.K. to sitcom The Big Bang Theory. What makes these works more humorous to the average person than similar ones? A good place to start might be with the show *@midnight* and their nightly *Hashtag Wars* segment. Each night viewers are given a prompt in the form of a Twitter hashtag and asked to tweet their funniest responses. Given two such tweets, how can we decide which is funnier?

This paper largely focuses on semantic relatedness-based features and their application in humour recognition. It is reasonable to assume that a punchline should be related to a setup; that is to say, a tweet's relevance to its hashtag prompt should be apparent. Similarly, it is reasonable to assume that punchlines should have a certain amount of unexpectedness; in other words, funnier tweets should be harder to guess. As such, it follows that semantic relation strength in general should serve as a barometer for humour where weaker relations are less understandable and stronger relations are more obvious (Cattle and Ma, 2016). Moreover, we hypothesize that the interplay between this understandability and unexpectedness should provide an even more powerful indication of humour.

2 Previous Work

Early work on computational humour focused more on humour generation in specific contexts, such as punning riddles (Binsted and Ritchie, 1994; Ritchie et al., 2007), humorous acronyms (Stock and Strapparava, 2003), or jokes in the form of "I like my X like I like my Y" (Petrovic and Matthews, 2013). Labutov and Lipson (2012) offered a slightly more generalized approach using Semantic Script Theory of Humour.

Recently, humour recognition has gained increasing attention. Taylor and Mazlack (2004) presented a method for recognizing wordplay in "Knock Knock" jokes. Mihalcea and Strapparava (2005) used stylistic features, such as alliteration and antonymy, to identify humorous one-liners. Mihalcea and Pulman (2007) expanded on this approach, finding that human-centeredness and negative sentiment are both useful in not only identifying humorous one-liners, but also distinguishing satirical news articles from genuine ones. Related to humor recognition, irony identification (Davidov et al., 2010; Tsur et al., 2010; Reyes et al., 2012) typically uses n-gram and sentiment fea-

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 401–406,
Vancouver, Canada, August 3 - 4, 2017.

tures to distinguish ironic from non-ironic tweets.

Shahaf et al. (2015) and Radev et al. (2016) examine humour recognition as a ranking task. Both works aim to identify the funnier of a pair of cartoon captions taken from submissions to The New Yorker's Cartoon Caption Contest[1]. Each week, New Yorker readers are presented "a cartoon in need of a caption" and encouraged to submit their own humorous suggestions. Shahaf et al. (2015) found that simpler grammatical structures, less reliance on proper nouns, and shorter joke phrases all lead to funnier captions. Radev et al. (2016) showed that in addition to human-centeredness and sentiment, high LexRank score was a strong indication of humour, where LexRank is a graph-based text summarization technique introduced in Erkan and Radev (2004).

Cattle and Ma (2016) noted that cartoon caption contests and hashtag wars are very similar in that they both involve short, humorous texts written as a response to an external stimulus. Furthermore, Cattle and Ma (2016) explored the role of semantic relatedness between setups and punchlines in perceived humour and found USF Free Association Norm- (Nelson et al., 2004) and Normalized Google Distance-based features (Cilibrasi and Vitanyi, 2007) to be useful in identifying funnier tweets. However, the results of Cattle and Ma (2016) were based on a small dataset of only four hashtag prompts, inferred humour judgments from Twitter likes and retweets, and relied on human annotations to identify both setups and punchlines.

3 System Definition

3.1 Dataset

We performed all training and testing on the dataset introduced in Potash et al. (2017) specifically for this task. The dataset consists of response tweets to 112 hashtags created by @midnight. The tweets are separated into files according to their respective hashtags, each hashtag file containing an average of 114 tweets. Each tweet includes a label specifying whether it was deemed to be funniest, in the top ten, or neither for that particular hashtag according to the @midnight staff. Potash et al. (2017) further divides the hashtags into three sets: Trial, Training, and Evaluation containing five, 101, and six hashtags, respectively.

[1] http://contest.newyorker.com/

3.2 Preprocessing and Baseline Features

Before feature extraction, tweets went through a preprocessing procedure. Each tweet was lowercased and then tokenized and POS tagged using Tweet NLP (Gimpel et al., 2011; Owoputi et al., 2013). English stop words were removed along with any punctuation, discourse markers (e.g. "RT"), interjections, emoticons, and URLs according to Tweet NLP's POS tags. Furthermore, prepositions, postpositions, subordinating conjunctions, coordinating conjunctions, verb particles, and predeterminers were also removed as these tended to be closed-class words (Gimpel et al., 2011) which do not affect the word-level semantic relationships this paper focuses on. Any references to the *@midnight* Twitter account or the relevant hashtag prompt were also removed. Each hashtag prompt was tokenized according to the hashtag segmentations included with the dataset. English stop words were removed along with any single digit numbers and the words "in" and "words". This was to omit the collocation "in # words", a common type of hashtag prompt, which does not affect semantic meaning.

Following the model of Shahaf et al. (2015), unigram and bigram bag-of-words features were extracted for each tweet. Furthermore, both unigram and bigram perplexities were calculated based on a simple language model created using n-gram counts from the Rovereto Twitter N-Gram Corpus (Herdağdelen, 2013). For simplicity, our language model uses add-one smoothing, although we intend to explore more complex smoothing techniques in future works. These features were intended to serve as a baseline for semantic relatedness-based features.

3.3 Semantic Relatedness Features

The results of Cattle and Ma (2016) suggest that University of Southern Florida Free Association Norm-based features are useful in humour recognition. Given two words, A and B, the USF Free Association Norms (USF) report the *forward strength*, i.e. proportion of participants who, when given word A, produce word B as their first reaction (Nelson et al., 2004). The USF dataset was represented as a graph where each node U referred to a word in the vocabulary and each edge from U to V had a weight proportional to the negative log of the forward strength form word U to word V. By representing the forward strengths as their

negative logs, finding the shortest path between two nodes using Dijkstra's Algorithm is equivalent to finding the path with the maximal product of forward strengths. Using this information we can easily estimate the forward strength between any two words in the USF vocabulary.

Word association is unidirectional; given the word "beer" a participant might say "glass" but given "glass" they might not say "beer" (Ma, 2013). Thus, we collect both $\text{USF}_{\text{forward}}$, representing how strongly the words in the hashtag prompt are associated with the words in the tweet's content, and $\text{USF}_{\text{backward}}$, representing how strongly the tweet's content is associated with the hashtag. These can be roughly interpreted as how easy a punchline is to guess given only the setup and how easy a punchline is to understand in context, respectively. Unlike Cattle and Ma (2016), which used human annotations to limit their scope to only punch words, we consider all hashtag-word/tweet-word pairs. We record the maximum, minimum, and average values for each feature across all such pairs.

Since we expect tweets which are relatively unexpected, i.e. low $\text{USF}_{\text{forward}}$, but also relatively easy to understand, i.e. high $\text{USF}_{\text{backward}}$, to be deemed funnier, we also collect $\text{USF}_{\text{difference}}$, the difference between the two values. $\text{USF}_{\text{difference}}$ is calculated both at word-level, e.g. $\text{USF}^{\text{word-max}}_{\text{difference}}$ refers to the maximal difference for a single word, and tweet-level, e.g. $\text{USF}^{\text{tweet-max}}_{\text{difference}}$ refers to the difference between $\text{USF}^{\text{max}}_{\text{forward}}$ and $\text{USF}^{\text{max}}_{\text{backward}}$.

In addition to USF association-based features, we also extract an identical set of features in the same manner but using the association strengths reported in the Edinburgh Associative Thesaurus (EAT) (Kiss et al., 1973).

To test the effectiveness of association-based features, we also collected the maximum, minimum, and average Word2Vec (Mikolov et al., 2013) cosine similarities across all hashtag-word/tweet-word pairs to serve as a semantic-feature baseline. We used Google's pre-trained Word2Vec embeddings[2].

3.4 Classifier

Features were extracted for each tweet following the methodology presented in the previous sections. Next, for each hashtag, tweet pairs were generated such that the two tweets had different humour judgment labels, i.e. one of the tweets is judged funnier according to the gold standard ratings. Each tweet pair then became two ordered training examples; one where the funnier of the two tweets was on the left and one where the funnier tweet was on the right, with appropriate training labels. For each training example, the left tweet's feature vector was concatenated with that of the right tweet's as well as the difference between the two. These training vectors were then used to train a Random Forest Classifier using scikit-learn[3], a popular Python machine learning library, using default settings and 100 estimators.

4 Results and Discussion

Feature selection experiments were performed using Training data for training and Trial data as a validation set to identify the best performing features. Using these features, we trained a new classifier on a combination of Training and Trial data and evaluated its performance on Evaluation data. The results in Table 1 show that the highest performing features in the validation test were $\text{EAT}^{\text{tweet-avg}}_{\text{difference}}$, $\text{USF}^{\text{max}}_{\text{forward}}$, $\text{EAT}^{\text{word-avg}}_{\text{difference}}$, $\text{USF}^{\text{word-avg}}_{\text{difference}}$, $\text{EAT}^{\text{min}}_{\text{forward}}$, $\text{USF}^{\text{tweet-max}}_{\text{difference}}$, and $\text{EAT}^{\text{min}}_{\text{backward}}$. The results using only these features are reported as **Best Features**. We also evaluated the performance of two more feature combinations: best features plus perplexity and n-gram features, as **Best Features+**, and best feature plus perplexity features only, as **Best Features+ (no n-gram)**.

Interestingly, although n-gram features on their own performed no better than chance in both validation and evaluation tests, their addition to the Best Features resulted in a large 8% point gain in validation tests compared to the same features minus n-grams (p=0.02 for paired t-test on file-level accuracies). However, their addition resulted in a drop in performance in evaluation tests, although this result was not statistically significant. Considering the dataset contains under 13,000 unique tweets, this extreme variation in performance might be due to n-gram features overfitting on the small dataset. By comparison, Shahaf et al. (2015) found n-grams alone offered a 55% accuracy on the similar task of selecting the funnier of two cartoon captions but their dataset contained over four times as many unique documents.

Another problem facing n-gram features is that

[2]https://code.google.com/archive/p/word2vec/

[3]http://scikit-learn.org/

	Features	Accuracy % Trial	Evaluation
baseline	n-grams	50.42	50.20
	unigram perplexity	53.05	50.27
	bigram perplexity	54.29	53.78
	Word2Vec Sim	50.72	50.76
Best Features	$EAT^{tweet\text{-}avg}_{difference}$	57.40	46.54
	$USF^{max}_{forward}$	55.15	48.33
	$EAT^{word\text{-}avg}_{difference}$	54.80	46.18
	$USF^{word\text{-}avg}_{difference}$	54.80	51.63
	$EAT^{min}_{forward}$	53.84	47.29
	$USF^{tweet\text{-}max}_{difference}$	52.44	50.58
	$EAT^{min}_{backward}$	51.94	44.33
	Best Features	**53.42**	**52.40**
	Best Features+	**61.51**	**53.72**
	Best Features+ (no n-gram)	**53.39**	**56.42**

Table 1: Accuracy by feature on Trial and Evaluation data

compared to cartoon captions, hashtag wars have a higher incidence of novel word-forms, typically in service of a pun, which occur only a few times for a particular hashtag prompt and never again. E.g. "HELLMFAO. #SpookyBands @midnight", or "Purrassic Park #CatBooks @Midnight". Simple n-gram models, such as the one used in this paper, are ill-equipped to deal with these types of out-of-vocabulary words.

Compared to Word2Vec, association-based features proved more discerning. One possible explanation for this is that word association is a more flexible relatedness measure than similarity. It is hard to find examples of similar concepts which are not also associated, but easy to find examples of associated concepts which are not similar. E.g. "red" and "green" would be similar in that they are both colours and associated in that they appear together at Christmas. However, "green" and "grass" are associated in that grass is green but the two words are very different.

Another possible explanation is that word associations are unidirectional while most similarity or distance metrics are not. The fact that four of the top seven best features are some variation of association difference seems to support our hypothesis that the interplay between a joke's unexpectedness and its understandability serves as a useful indication of humour.

Cattle and Ma (2016) noted that their USF performance was hurt by a lack of coverage. This seems to be the case for us as well. Less than 65% of tweets contained a valid $USF_{forward}$ strength, with valid $USF_{backward}$ and $USF_{difference}$ strengths appearing in only 70% and less than 60%, respectively. By comparison, almost 95% of tweets contained a valid Word2Vec similarity. EAT showed slightly better coverage with valid $EAT_{forward}$ and $EAT_{backward}$ strengths each appearing in almost 75% of tweets and $EAT_{difference}$ appearing in just under 70%. This may explain why EAT features outperformed USF in the validation tests. This was expected given that EAT contains twice as many words as USF and four times as many edges.

In order to avoid using human annotations, such as those in Cattle and Ma (2016), USF, EAT, and Word2Vec features were calculated across all hashtag-word/tweet-word pairs. Even though the bag-of-words was heavily filtered by POS to leave only words which carry more word-level semantic meaning, this is a shotgun-like approach which likely added noise to the data. This may explain why only two of the top seven word association features use min values. The punchline makes up only a small part of the tweet and it is expected that the remainder would not show any strong associations. Some kind of automatic punchline identification would help in this respect but may exacerbate the aforementioned coverage issue faced by USF and EAT.

5 Conclusion and Future Work

Humour recognition in general is a very difficult task and humour recognition in hashtag wars is no exception. The majority of features tested performed only slightly better than chance if better than chance at all. Our optimal result was only a 56.42% accuracy on Evaluation data and was obtained using only the features unigram perplexity, bigram perplexity, $EAT^{tweet\text{-}avg}_{difference}$, $USF^{max}_{forward}$, $EAT^{word\text{-}avg}_{difference}$, $USF^{word\text{-}avg}_{difference}$, $EAT^{min}_{forward}$, $USF^{tweet\text{-}max}_{difference}$, and $EAT^{min}_{backward}$. Although our accuracy is fairly low we believe semantic relatedness features, and word association-based features in particular, are worthy of further study.

Our system could be improved by using automatic punchline detection to cut down on the noise in our word association features. Furthermore, a larger vocabulary or even the ability to automatically infer association strength would increase the usefulness of word association features. Finally, a larger dataset may be needed to rule out the efficacy of n-gram features.

References

Kim Binsted and Graeme Ritchie. 1994. An implemented model of punning riddles. In *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*. AAAI Press, pages 633–638.

Andrew Cattle and Xiaojuan Ma. 2016. Effects of semantic relatedness between setups and punchlines in twitter hashtag games. *PEOPLES 2016* page 70.

Rudi L Cilibrasi and Paul MB Vitanyi. 2007. The google similarity distance. *IEEE Transactions on knowledge and data engineering* 19(3).

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*. Association for Computational Linguistics, pages 107–116.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22:457–479.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pages 42–47.

Amaç Herdağdelen. 2013. Twitter n-gram corpus with demographic metadata. *Language resources and evaluation* 47(4):1127–1147.

George R Kiss, Christine Armstrong, Robert Milroy, and James Piper. 1973. An associative thesaurus of english and its computer analysis. *The computer and literary studies* pages 153–165.

Igor Labutov and Hod Lipson. 2012. Humor as circuits in semantic networks. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, pages 150–155.

Xiaojuan Ma. 2013. Evocation: analyzing and propagating a semantic link based on free word association. *Language resources and evaluation* 47(3):819–837.

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*. Springer, pages 337–347.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 531–538.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers* 36(3):402–407.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Sasa Petrovic and David Matthews. 2013. Unsupervised joke generation from big data. In *ACL (2)*. Citeseer, pages 228–232.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Semeval-2017 task 6: #hashtagwars: Learning a sense of humor. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 49–57. http://www.aclweb.org/anthology/S17-2004.

Dragomir Radev, Amanda Stent, Joel Tetreault, Aasish Pappu, Aikaterini Iliakopoulou, Agustin Chanfreau, Paloma de Juan, Jordi Vallmitjana, Alejandro Jaimes, Rahul Jha, and Robert Mankoff. 2016. Humor in collective discourse: Unsupervised funniness detection in the new yorker cartoon caption contest. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering* 74:1–12.

Graeme Ritchie, Ruli Manurung, Helen Pain, Annalu Waller, Rolf Black, and Dave OMara. 2007. A practical application of computational humour. In *Proceedings of the 4th International Joint Conference on Computational Creativity*. pages 91–98.

Dafna Shahaf, Eric Horvitz, and Robert Mankoff. 2015. Inside jokes: Identifying humorous cartoon captions. In *Proceedings of the 21th*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, KDD '15, pages 1065–1074. https://doi.org/10.1145/2783258.2783388.

Oliviero Stock and Carlo Strapparava. 2003. Hahacronym: Humorous agents for humorous acronyms. *Humor* 16(3):297–314.

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

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*. pages 162–169.