Recognizing Humour using Word Associations and Humour Anchor Extraction

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 attempts to marry the interpretability of statistical machine learning approaches with the more robust models of joke structure and joke semantics capable of being learned by neural models. Specifically, we explore the use of semantic relatedness features based on word associations, rather than the more common Word2Vec similarity, on a binary humour identification task and identify several factors that make word associations a better fit for humour. We also explore the effects of using joke structure, in the form of humour anchors (Yang et al., 2015), for improving the performance of semantic features and show that, while an intriguing idea, humour anchors contain several pitfalls that can hurt performance.

1 Introduction

Humour is a part of everyday communication. Although telling and understanding jokes comes naturally to most humans, the recognition and interpretation of humour is a very difficult task for computers. Beyond merely expressing a funny idea, great jokes choose evocative words and present them in a surprising structure (Attardo, 2008) often with pleasing phonetics (Mihalcea and Strapparava, 2005). In addition to this linguistic mastery, many jokes require world knowledge either by setting up and then subverting an expectation or by referencing a common piece of culture (Kukovačec et al., 2017). This makes computational humour a challenging yet very intriguing problem for natural language processing (NLP). As such, it is no surprise that computational humour, and humour recognition in particular, has received increased attention from the NLP community with SemEval-2017 devoting two tasks to it: ranking humorous tweets (Potash et al., 2017) and interpreting English puns (Miller et al., 2017).

This recent attention has lead to advancements such as sequence-based neural humour models capable of implicitly learning a joke structure and semantic features (Bertero and Fung, 2016a; Donahue et al., 2017). While these approaches offer good performance, their reliance on complicated neural architectures over explicitly engineered features present a problem for interpretability which may make it difficult to diagnose problems if results go wrong.

Works which take a more interpretable statistical machine learning approach have their own drawbacks. For example, the representation of joke semantics has been fairly basic, typically computing word embedding similarities between all word pairs in a document (Yang et al., 2015), and bear little resemblance to the way humans actually interpret humour. Additionally, many works fail to take advantage of joke structure, treating texts as unordered bags-of-words (Bertero and Fung, 2016b; Mihalcea and Strapparava, 2005; Yan and Pedersen, 2017).

This paper aims to marry the interpretability of statistical machine learning approaches with the more nuanced models of joke structure and joke semantics of neural approaches. Specifically, we explore the effectiveness modelling joke semantics using semantic relatedness features based on word associations, rather than the more common semantic similarity features based on word embeddings. We present evidence not only that relatedness in general is better suited than similarity for computational humour tasks

Proceedings of the 27th International Conference on Computational Linguistics, pages 1849–1858
Santa Fe, New Mexico, USA, August 20-26, 2018.

due to its broader nature, but also that word associations are particularly well suited due to their ability to map more nuanced relationships (De Deyne and Storms, 2008) and asymmetric nature. While such features have been explored in the past (Cattle and Ma, 2016; Cattle and Ma, 2017b), this work presents a more in depth analysis, focusing on a more fundamental task (binary humour classification vs. relative humour ranking) on with a dataset that better represents natural language (oneliners and puns vs. Twitter hashtag games), and is the first work to incorporate interpolated word association strengths. Furthermore, we introduce a novel method for targetting our semantic features using joke structure to help reduce noise and increase reliability. Specifically, we experiment with integrating the extraction of humour anchors, the "meaningful, complete, minimal set of word spans" (Yang et al., 2015) that allow humour to occur, into the humour classification process itself, the first work to do so. We are also making the code used to run these experiments publicly available[1].

2 Related Work

Informally, jokes are generally divided into setups and punchlines, with the setup establishing a context and the punchline delivering the humour. More formally, according to the Semantic Script Theory of Humor (SSTH) introduced in Raskin (1985), humour is derived from the resolution of two overlapping, but incongruent scripts of differing levels of obviousness (Labutov and Lipson, 2012). That is to say, humour comes from causing listeners to make assumptions about a situation (i.e. implying the obvious script) and then subverting them (i.e. forcing a re-evaluation which results in the choosing of the less obvious one). Consider the following joke taken from Raskin (1985):

> "Is the doctor at home?" the patient asked in his bronchial whisper. "No," the doctor's young and pretty wife whispered in reply. "Come right in."

The use of the words *doctor*, *patient*, and *bronchial* in the setup lead the listener to assume that the man is seeking medical advice. However, the punchline, the doctor's wife's response along with her description, reveals the man's true intent.

The General Theory of Verbal Humor (GTVH), introduced in Attardo and Raskin (1991), expanded on SSTH, retaining the notion of script opposition but highlighting other factors which also affect humour. These include "target (i.e., the person or group made fun of in the joke), the situation (i.e., the background assumed by the joke, such as the props for the story), the narrative strategy (the 'genre' of the joke), and the language (i.e., the lexical, syntactic choices of the text)." (Attardo, 2008)

While target, situation, and the overlapping but incongruent nature of the scripts speak to the importance of semantics in humour, narrative strategy, language, and the fact that listeners should pick the more obvious script first speak to the importance of structure. Although these two aspects are not entirely independent, for the purposes of this paper, it is useful to consider them separately.

2.1 Joke Semantics

Following SSTH, we expect that punchlines that do not sufficiently overlap with their setup are unfunny as they do not flow logically from the context. Similarly, punchlines that are not sufficiently incongruent with their setup are unfunny as there is no re-evaluation. As overlap and incongruity are difficult to measure directly, one common approach is instead to use word embeddings, such as Word2Vec (Mikolov et al., 2013), to calculate the cosine similarities between pairs of vectors representing words in a document (Yang et al., 2015; Shahaf et al., 2015; Kukovačec et al., 2017). However, measuring incongruity and overlap in terms of similarity is a rather odd choice. Just because two scripts overlap does not imply they are similar. In the "doctor" example in Section 2, the two scripts, namely *[the patient is seeking medical advice]* and *[the patient is having an affair with the doctor's wife]*, overlap in terms of the people and locations involved but otherwise are quite different. Similarly, incongruent scripts such as *[dog bites man]* and *[man bites dog]* are very similar in all but the assignment of the roles.

Compared with *similarity*, *relatedness* is a much broader concept. It is easy to think of concepts that are related but not similar. For example, "beer" and "glass" are not similar, describing very different

[1] `https://github.com/acattle/HumourTools`

concepts, but are quite closely related in that beer often comes in glasses (Ma, 2013). However, it is difficult to think of two concepts which are similar but not related. Although similarity is much more common, several semantic relatedness measures do exist. Extended Lesk (Banerjee and Pedersen, 2003), for example, compute the relatedness between two words as a function of the size of the overlap of their glosses. However, such measures do not fully address the short comings of word embedding similarities.

Any measure based on a distributional semantic approach, including both Extended Lesk and Word2Vec, at its core relies on word co-occurrence to extract relationships. However, not all relationships are evidenced by co-occurrence. For example, “yellow” and “banana” are so closely related that “yellow banana” occurs relatively infrequently since bananas are assumed to be yellow unless told otherwise (De Deyne et al., 2016).

Word associations capture relationships between concepts, are not reliant on distributional semantics, and have long been used in psychology and cognitive science dating back to Galton (1879). Furthermore, some word association-based metrics have been shown to perform better than Word2Vec on a series of similarity tasks (De Deyne et al., 2016). Although the collection of word association datasets is much more labour intensive than distributional semantic approaches, several word association datasets do exist, the most popular being the Edinburgh Associative Thesaurus (EAT), collected in Kiss et al. (1973), and the University of South Florida Free Association Norms (USF), collected in Nelson et al. (2004). Both these datasets were compiled by presenting participants with a cue word and asking them to respond with the first word that comes to mind. The proportion of respondents who give a specific response for a specific cue is called the *forward strength* from the cue to the target. This approach does have its drawbacks, forward strengths are relative instead of absolute (Nelson et al., 2004) and accepting only a single response causes weaker relationships to be under-represented (De Deyne and Storms, 2008), but it is a straightforward and widely accepted way of collecting word association data.

This not the first work to propose word associations for humour recognition. Cattle and Ma (2016) and its follow up, Cattle and Ma (2017b), previously explored the effectiveness of word association strengths on a relative humour ranking task with encouraging results. Most notably, they examined the role of word association asymmetry. While the cosine similarity function preferred by vector space models like Word2Vec and TFIDF is symmetrical, word associations are not. Someone shown the word “beer” may be likely to say “glass” but someone shown the word “glass” may be unlikely to say “beer” (Ma, 2013). This allows for finer grained exploration of the relationship between setups and punchlines. In fact, Cattle and Ma (2016) shows evidence that punchlines tend to be funnier if the strength from the setup to the punchline, which they claim represents how obvious a punchline is, is weaker than the strength of the punchline back to the setup, which they claim represents how easy a punchline is to understand.

While our approach is similar to that of Cattle and Ma (2017b), we note several key differences. First, they perform a relative humour ranking task as opposed to our binary humour classification task. As relative ranking is a more difficult task in general (Potash et al., 2017), Cattle and Ma (2017b)’s results were far from convincing. By pursuing the easier binary classification task, we intend to present a clearer evaluation of word association features for humour recognition. Second, Cattle and Ma (2017b) focused on Twitter hashtag games, a type of online game where participants submit their best responses to a central topic or theme. Compared to the oneliners and puns explored in this work, not only are hashtag war entries slightly shorter, but hashtag wars also include a clear and explicitly marked setup in the form of the central hashtag (Cattle and Ma, 2016). Third, Cattle and Ma (2017b) exclusively utilize a graph-based method for extracting association strengths. This led to coverage issues due to the relatively small vocabularies of the word association datasets. We address this issue by using an machine learning-based approach capable of predicting strengths between arbitrary word pairs, similar to the one introduced in (Cattle and Ma, 2017a).

The association strength prediction method introduced in Cattle and Ma (2017a) is in turn very similar to the WordNet Evocation (Boyd-Graber et al., 2006) prediction method introduced in Hayashi (2016). In all cases, association strengths are estimated using a variety of WordNet (Miller and Fellbaum, 1998) and vector space features which are fed into a multi-layer perceptron.

2.2 Joke Structure

Early humour generation systems tended to model jokes' setup/punchline structure explicitly. HAHAcronyms (Stock and Strapparava, 2002) takes well-known acronyms as setups and, using listener's preconceived notions of that acronym's canonical expansion, generates novel expansions as punchlines which present an ironic clash. Petrović and Matthews (2013) generate jokes of the form "I like my *X* like I like my *Y*, *Z*," where the punchline *Z* acts as a link between the setups *X* and *Y*. Labutov and Lipson (2012) invokes SSTH directly, generating setups by mixing compatible elements of two overlapping but incongruent scripts of varying degrees of obviousness, generated using the common-sense knowledgebase ConceptNet (Speer and Havasi, 2012), and punchlines which introduce information compatible with only the second, less obvious script.

The punning riddle generators, such as Binsted and Ritchie (1994), present an rather unique version of setups and punchlines. Considering the example "What do you call a *sour assistant*? A *lemon aide*." (Binsted, 1996), the setup, instead of establishing an expectation to be subverted, actually establishes a context in which the somewhat odd reading of *lemon aide* can be preferred over the much more common and phonetically identical reading *lemonade*. However, it should be noted that clearly defined setups and punchlines still exist.

Given humour generation's firm grasp of punchline/setup structure, it is somewhat surprising that humour recognition has largely ignored it despite humour recognition starting in earnest with Mihalcea and Strapparava (2005), a full decade after early humour generation works such as Binsted and Ritchie (1994). Early humour recognition works starting with Mihalcea and Strapparava (2005) and its follow-up Mihalcea and Pulman (2007), up to more recent works such as Radev et al. (2016) and Yan and Pedersen (2017) employ bag-of-words models. Such systems are unable to capture document-level structure, such as that setups tend to precede punchlines.

As mentioned in Section 1, joke structure and joke semantics are not entirely independent. This means poor models of joke structure can affect the performance of features designed to capture joke semantics. Despite the issues mentioned in Section 2.1, Yang et al. (2015)'s "incongruity" feature set, maximum and minimum word embedding similarities between pairs of words in a document, perform fairly well. Cattle and Ma (2017b) takes a similar approach with their word association features. The problems comes from the fact that both works compute these values across *all* possible pairs of words in a document. This can introduce noise as not all word pairs are meaningful (e.g. pairs of stopwords) and internally-cohesive setups and punchlines can bias maximum similarity scores. While this can be somewhat alleviated by judicial filtering of stopwords, this does not guarantee meaningful word pairs either.

Yang et al. (2015), in addition to their humour classifier, also introduces a method for identifying jokes' humour anchors (HAs), the "meaningful, complete, minimal set of word spans" that allow humour to occur. While this is slightly different from identifying a joke's setup and punchline, focusing only on pairs of HAs would help reducing noise by increasing the precision of meaningful word pairs selection without sacrificing recall. However, Yang et al. (2015) does not use their extracted HAs to improve their humour classification performance. Likely this is due to the fact that their proposed HA extraction method requires a separate, fully trained humour prediction model which is robust to word order. Furthermore, the quality of the extracted HAs is directly tied to the quality of the humour model. However, both issues could have been alleviated using some form of co-training or bootstrapping between the overall humour prediction model and the HA extractor's internal humour prediction model. The HA extractor works by generating a list of HA candidates for each document following a heuristic method. After removing various combinations of HA candidates from the original document, these modified documents are fed into a trained humour prediction model, with the HAs being the combination of HA candidates which causes the largest drop in humour score. Since the humour prediction model is by design robust against word order, words can be freely omitted with few side effects.

It should be noted that sequence-based humour models such as Bertero and Fung (2016b) or Donahue et al. (2017) should theoretically be capable of implicitly learning HAs, especially Bertero and Fung (2016a)'s Long Short-Term Memory-based approach. However, these models are much more complex than Yang et al. (2015)'s approach, require more training data, and suffer from a lack of interpretability.

3 Methodology

3.1 Datasets

We evaluate our classifiers across two separate datasets: Pun of the Day (PotD), collected in Yang et al. (2015), and 16000 One-Liner (OL), collected in Mihalcea and Strapparava (2005). PotD consists of positive examples collected from the Pun of the Day website[2] and negative examples collected from a combination of news sources, question/answer forums, and lists of proverbs (Yang et al., 2015). OL consists of positive examples scraped from humour websites and negative examples taken from a combination of new headlines, sentences from the British National Corpus, and proverbs.

3.2 Baseline

For our baseline we implemented our own version of Yang et al. (2015)'s highest performing classifier. This model was chosen due to its high performance and its use of statistical machine learning techniques which make it a fair point of comparison. Features include, for each document, minimum and maximum Word2Vec similarities between all word pairs, the total number of word sense combinations in each document according to WordNet, minimum and maximum WordNet path similarities between all word pairs, number of words with negative/positive polarity as well as weak/strong subjectivities according to the Wilson et al. (2005) sentiment lexicon, number of and length of longest alliteration and rhyme chains according to the CMU Pronouncing Dictionary[3], labels of the five nearest neighbours in the training set according to word frequencies, and an averaged Word2Vec embedding across all words for a total of 318 feature dimensions. We then train a Random Forest Classifier using the scikit-learn[4] Python library with 100 estimators but otherwise default settings.

All Word2Vec features, including those described below, use Google's pre-trained 300 dimension Word2Vec embeddings[5].

3.3 Semantic Features

Similar to Yang et al. (2015), we compute the minimum, maximum, and average Word2Vec similarity between ordered word pairs. Not only is this a common humour recognition feature (Yang et al., 2015; Shahaf et al., 2015; Kukovačec et al., 2017), but it also acts as a point of comparison for word association strength.

For our word association features we compute the minimum, maximum, and average association strength between ordered word pairs, which we refer to this as the *forward* strength. Since, as described in Section 2.1, word associations are directional, we also compute the minimum, maximum, and average associations strengths between the reverse ordered word pairs, which we refer to as the *backward* strength. Following Cattle and Ma (2016), we also compute the difference between these two values on both a micro (per word) and macro (per document) level. We refer to these sets of features as the *diff* strengths. Forward, backward, and diff strengths are extracted for both the EAT and USF word association datasets.

We also compare two methods for computing our word association strength features. The first method uses the graph-based method described in Cattle and Ma (2016). We refer to this approach as *graph*. The second uses an in-house association strength predictor using an machine learning-based method similar to the one described in Cattle and Ma (2017a). We refer to this approach as *ML*. Unlike Cattle and Ma (2017a), which uses three different types of word embeddings, we use only Word2Vec. In addition to Word2Vec similarity and vector offsets (the difference between the cue and response vectors), we also include LDA similarity (300 dimensions, trained using Gensim[6] on English Wikipedia), AutoExtend (Rothe and Schütze, 2015) similarity, and a variety of WordNet-based features described in Cattle and Ma (2017a).

[2] http://punoftheday.com/

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

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

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

[6] https://radimrehurek.com/gensim/

3.4 Humour Classifier

In addition to the semantic features described above, we also calculate each document's perplexity according to a 3gram language model (LM) trained on the WMT15 English news discussion corpus (Bojar et al., 2015) using KenLM[7]. This feature is included not only because document perplexity has been shown to be useful for humour recognition (Shahaf et al., 2015) but also to act as a baseline for our semantic features. Because both the datasets described in Section 3.1 draw negative examples from news sources, we were concerned that training an LM on a general English corpus might unfairly bias the perplexity scores against those negative examples. Similarly, training an LM on a news corpus might unfairly bias perplexity towards those negative examples. Therefore, the news discussion corpus was chosen as a happy medium between news vocabulary and informal writing styles.

As with our baseline, our extracted features are used to train a Random Forest Classifier using scikit-learn with 100 estimators and default settings using a 10 fold cross validation.

3.5 Humour Anchors

By default, word association and Word2Vec features are computed across all word pairs in a document. However, we also experiment computing these features only across pairs of humour anchor (HA) words. HAs are extracted using the method described in Yang et al. (2015) using the same baseline humour model described in Section 3.2 for anchor candidate evaluation.

HA extraction's requirement of a fully trained anchor candidate evaluator raises the problem of what data that evaluator should be trained on. Given that, as described in Section 3.1, we experiment on two separate humour datasets, we train the anchor candidate evaluator on the opposite dataset from the overall humour classifier. This is to avoid overfitting or biasing the anchor candidate evaluator by training on the test data.

4 Results and Discussion

The results of our experiments are reported in Table 1. In general, our model performs slightly worse than the Yang et al. (2015) baseline. One interesting aspect to note is that our model uses only 28 feature dimensions compared to Yang et al. (2015)'s 318. While this is not exactly a fair comparison in the case of our ML-based word association strengths (our ML strength predictor takes 415 feature dimensions as input), graph-based associations perform similarly and do truly use only 28 dimensions. Overall performance is similar across both datasets with the only notable exception being graph-based USF performing better on OL than PotD. This is likely due to OL being better suited than PotD to USF's relatively smaller set of associations (72,176 pairs and 10,617 unique words versus EAT's 325,588 and 23,218).

4.1 Word Association Strengths

As shown in Table 1, word association strength features outperform Word2Vec similarity. This provides the first clear evidence of their usefulness in humour recognition. This is particularly notable in the case of graph-based associations as they are based on a much smaller dataset and vocabulary.

Despite differences in their individual feature performance on PotD, our model's overall performance is very similar for both graph- and ML-based word association features. This was somewhat unexpected as ML-based associations were included specifically to address coverage issues with graph-based associations noted in Cattle and Ma (2017b). However, graph-based associations seem to exhibit an acceptable level of performance on both datasets, even outperforming ML on OL. One possible explanation is document length. Tweets in Potash et al. (2017)'s dataset averaged 5.5 words (after removing each game's hashtag and the source TV show's Twitter username) versus 13.3 for PotD puns and 10.6 for OL one-liners. As document length increases, the number of word pairs increases exponentially, raising the likelihood of finding at least one word pair with a valid word association strength.

It was expected that ML-based associations would outperform Word2Vec similarity since, as noted in Section 3.3, our association strength prediction model uses Word2Vec similarity as an input. However,

[7] `https://kheafield.com/code/kenlm/`

	Pun of the Day				**16000 One-Liners**			
	Acc	P	R	F1	Acc	P	R	F1
Yang et al. (2015)	0.795	0.761	0.862	0.808	0.798	0.801	0.794	0.797
All Features (graph)	0.757	0.755	0.764	0.759	**0.759**	**0.745**	**0.787**	**0.765**
All Features (ML)	**0.763**	**0.757**	**0.780**	**0.768**	0.742	0.727	0.777	0.751
All Features (ML) + HA	0.711	0.700	0.741	0.720	0.679	0.657	0.748	0.700
Perplexity	0.610	0.613	0.608	0.610	0.591	0.591	0.589	0.590
Word2Vec	0.658	0.652	0.685	0.668	0.657	0.655	0.661	0.658
Word Associations (graph)	0.695	0.692	0.706	0.699	**0.747**	**0.735**	**0.773**	**0.753**
Word Associations (ML)	**0.713**	**0.703**	**0.741**	**0.722**	0.720	0.712	0.740	0.726
EAT (graph)	0.670	0.665	0.691	0.678	0.729	0.716	0.761	0.738
EAT (ML)	0.697	0.690	0.722	0.706	0.708	0.701	0.725	0.713
EAT (ML) forward	0.665	0.664	0.674	0.669	0.668	0.667	0.673	0.670
EAT (ML) backward	0.641	0.640	0.651	0.646	0.662	0.663	0.661	0.662
EAT (ML) micro diff	0.563	0.568	0.541	0.555	0.612	0.607	0.637	0.622
EAT (ML) macro diff	0.557	0.558	0.567	0.562	0.606	0.613	0.575	0.594
USF (graph)	0.611	0.618	0.586	0.602	0.733	0.723	0.754	0.738
USF (ML)	0.692	0.686	0.713	0.699	0.709	0.700	0.732	0.715
USF (ML) forward	0.676	0.674	0.685	0.679	0.667	0.664	0.677	0.670
USF (ML) backward	0.637	0.637	0.646	0.641	0.672	0.668	0.681	0.674
USF (ML) micro diff	0.567	0.572	0.544	0.558	0.639	0.631	0.666	0.648
USF (ML) macro diff	0.554	0.554	0.576	0.565	0.603	0.611	0.567	0.588

Table 1: Selected Results of Binary Humour Classification Experiments

it should be noted that, like Cattle and Ma (2017a)'s model, the single highest performing feature was not Word2Vec similarity but vector offset (the difference between two word's Word2Vec embeddings). This further highlights the limitations of Word2Vec similarity described in Section 2.1 as our strength predictor is clearly learning something that cannot be captured by cosine similarity alone.

While Cattle and Ma (2016) provided evidence that the difference between the forward and backward word association strengths was more useful than forward or backward strengths alone, this is not the case here. Both micro and macro difference performed worse than either forward or backward. Furthermore, forward and backward perform similarly. This is likely due to Cattle and Ma (2016)'s use of Twitter hashtag wars, meaning setup and punchline were clearly marked. As such, forward strengths were guaranteed to represent associations from the setup to the punchline and backward strengths were guaranteed to be from the punchline back to the setup. In our approach, setups and punchlines were not explicitly defined. This meant it was difficult to know if a specific word pair represented setup to punchline, punchline to setup, or even setup to setup or punchline to punchline. Even using humour anchors (HAs) does not solve this problem since, as referenced in Section 2.2, identifying HAs is slightly different from identifying setups and punchlines. While labelling a word span as a setup or a punchline gives us some insight into its purpose in the joke (i.e. whether it is meant to establish context or to trigger a reframing, respectively), HAs do not include this information.

4.2 Humour Anchors

As mentioned in Section 2.2, using HAs for humour recognition is an appealing notion and would allow semantic features to be targeted only to meaningful word pairs, potentially increasing their effectiveness. The wonderfully simple extraction method described in Yang et al. (2015) only makes HAs more intriguing. Unfortunately, as can be seen in Table 1, HA targetting actually hurts the performance of our humour model.

One obvious suspect for this drop in performance is the quality of the extracted HAs, a sample of which is shown in Table 2. While the *Honeymoon* and *LASER* examples seem relatively accurate, many

I'll **[Marry]** You Tomorrow But Let's **[Honeymoon Tonight]**.
Caution! Do not look into **[LASER]** with **[remaining eye]**.
Newsflash! Dyslexic Christian **[sells]** soul to **[Santa]**.
[Dark] is faster than light, **[otherwise]** you would see it.
If there **[were]** a single **[word]** to **[describe]** me it would have to be profectionist.

Table 2: Sample extracted humour anchors

extracted anchors were either incomplete, as is the case with *Dark* and *Santa*, or nonsensical, like *profectionist*.

As described in Section 2.2, Yang et al. (2015)'s HA extraction algorithm requires a fully trained humour model, the accuracy of which undoubtedly affects the quality of the extracted HAs. For this reason we also experimented with training our anchor candidate scorer using the test data, to maximize its performance. While this approach is problematic, it does provide an upper bound for our HA extraction performance. While using such HA targetted models did result in increased humour classification performance ($ACC = 0.740$, $P = 0.735$, $R = 0.754$, $F1 = 0.744$ on PotD. $ACC = 0.706$, $P = 0.678$, $R = 0.783$, $F1 = 0.727$ on OL.), it still failed to exceed our non-HA models.

We chose our baseline Yang et al. (2015) humour classifier as our anchor candidate scorer for simplicity but their HA extraction algorithm is able to work with any humour recognition model so long as it is robust to word order and capable of generating a humour score (in our case, we used humour probability). Therefore, using a more accurate humour model may have led to better performance.

Another reason HAs may have hurt humour recognition performance is that it may be make non-humorous documents more humorous. Yang et al. (2015)'s method finds the combination of humour anchor candidates that cause the largest drop humour score, i.e. by design it selects a subset of words which positively effect humour scores. As such, HAs may be more likely to be judged as humorous even if the documents they are extracted from are not.

5 Conclusion and Future Work

In this paper we have presented a humour classification system based on word associations and shown performance, across two datasets, above state-of-the-art non-neural models. Furthermore, we show that word association features outperform Word2Vec similarity on this task, providing the first significant evidence that word associations are particularly well suited to computational humour tasks. Furthermore, we explored the effectiveness of humour anchors for humour recognition and found they actually hurt performance.

Possible next steps include exploring the usefulness of word association features on other computational humour tasks such as relative humour ranking or even humour generation. On a more fundamental level, there are still many aspects of word association features left to explore such as examining different datasets or even different strength metrics (e.g. overlap strength, the proportion of shared associations between two words).

References

Salvatore Attardo and Victor Raskin. 1991. Script theory revis(it)ed: Joke similarity and joke representation model. *Humor-International Journal of Humor Research*, 4(3-4):293–348.

Salvatore Attardo. 2008. Semantics and pragmatics of humor. *Language and Linguistics Compass*, 2(6):1203–1215.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 805–810, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dario Bertero and Pascale Fung. 2016a. A long short-term memory framework for predicting humor in dialogues. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016-Proceedings of the Conference*, pages 130–135.

Dario Bertero and Pascale Fung. 2016b. Predicting humor response in dialogues from TV sitcoms. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5780–5784.

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

Kim Binsted. 1996. *Machine humour: An implemented model of puns*. Ph.D. thesis, The University of Edinburgh.

Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina. 2015. Proceedings of the tenth workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

Jordan Boyd-Graber, Christiane Fellbaum, Daniel Osherson, and Robert Schapire. 2006. Adding dense, weighted connections to WordNet. In *Proceedings of the Third Global WordNet Meeting*, Jeju Island, Korea.

Andrew Cattle and Xiaojuan Ma. 2016. Effects of semantic relatedness between setups and punchlines in Twitter hashtag games. *Workshop on Computational Modeling of Peoples Opinions, Personality, and Emotions in Social Media (PEOPLES 2016)*, page 70.

Andrew Cattle and Xiaojuan Ma. 2017a. Predicting word association strengths. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1283–1288, Copenhagen, Denmark, September. Association for Computational Linguistics.

Andrew Cattle and Xiaojuan Ma. 2017b. SRHR at SemEval-2017 task 6: Word associations for humour recognition. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 400–405, Vancouver, Canada, August. Association for Computational Linguistics.

Simon De Deyne and Gert Storms. 2008. Word associations: Norms for 1,424 dutch words in a continuous task. *Behavior Research Methods*, 40(1):198–205.

Simon De Deyne, Amy Perfors, and J. Daniel Navarro. 2016. Predicting human similarity judgments with distributional models: The value of word associations. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1861–1870. The COLING 2016 Organizing Committee.

David Donahue, Alexey Romanov, and Anna Rumshisky. 2017. HumorHawk at SemEval-2017 task 6: Mixing meaning and sound for humor recognition. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 98–102, Vancouver, Canada, August. Association for Computational Linguistics.

Francis Galton. 1879. Psychometric experiments. *Brain*, 2(2):149–162.

Yoshihiko Hayashi. 2016. Predicting the evocation relation between lexicalized concepts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1657–1668, Osaka, Japan, December. The COLING 2016 Organizing Committee.

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.

Marin Kukovačec, Juraj Malenica, Ivan Mršić, Antonio Šajatović, Domagoj Alagić, and Jan Šnajder. 2017. TakeLab at SemEval-2017 task 6: #rankinghumorin4pages. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 395–399, Vancouver, Canada, August. Association for Computational Linguistics.

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*, pages 150–155. Association for Computational Linguistics.

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*, pages 337–347. Springer.

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*, pages 531–538. Association for Computational Linguistics.

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.

George Miller and Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press Cambridge.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68, Vancouver, Canada, August. Association for Computational Linguistics.

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.

Saša Petrović and David Matthews. 2013. Unsupervised joke generation from big data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–232, Sofia, Bulgaria, August. Association for Computational Linguistics.

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)*, pages 49–57, Vancouver, Canada, August. Association for Computational Linguistics.

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, 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)*, Paris, France, 05. European Language Resources Association (ELRA).

Victor Raskin. 1985. Semantic theory of humor. In *Semantic Mechanisms of Humor*. Springer.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China, July. Association for Computational Linguistics.

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*, KDD '15, pages 1065–1074, New York, NY, USA. ACM.

Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3679–3686, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1639.

Oliviero Stock and Carlo Strapparava. 2002. HAHAcronym: Humorous agents for humorous acronyms. *Stock, Oliviero, Carlo Strapparava, and Anton Nijholt. Eds*, pages 125–135.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Xinru Yan and Ted Pedersen. 2017. Duluth at semeval-2017 task 6: Language models in humor detection. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 384–388, Vancouver, Canada, August. Association for Computational Linguistics.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376, Lisbon, Portugal, September. Association for Computational Linguistics.