Uncertainty and Surprisal Jointly Deliver the Punchline: Exploiting Incongruity-Based Features for Humor Recognition

Yubo Xie, Junze Li, and Pearl Pu
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
{yubo.xie,junze.li,pearl.pu}@epfl.ch

Abstract

Humor recognition has been widely studied as a text classification problem using data-driven approaches. However, most existing work does not examine the actual joke mechanism to understand humor. We break down any joke into two distinct components: the set-up and the punchline, and further explore the special relationship between them. Inspired by the incongruity theory of humor, we model the set-up as the part developing semantic uncertainty, and the punchline disrupting audience expectations. With increasingly powerful language models, we were able to feed the set-up along with the punchline into the GPT-2 language model, and calculate the uncertainty and surprisal values of the jokes. By conducting experiments on the SemEval 2021 Task 7 dataset, we found that these two features have better capabilities of telling jokes from non-jokes, compared with existing baselines.

1 Introduction

One of the important aspects of computational humor is to develop computer programs capable of recognizing humor in text. Early work on humor recognition (Mihalcea and Strapparava, 2005) proposed heuristic-based humor-specific stylistic features, for example alliteration, antonymy, and adult slang. More recent work (Yang et al., 2015; Chen and Soo, 2018; Weller and Seppi, 2019) regarded the problem as a text classification task, and adopted statistical machine learning methods and neural networks to train models on humor datasets. However, only few of the deep learning methods have tried to establish a connection between humor recognition and humor theories. Thus, one research direction in humor recognition is to bridge the disciplines of linguistics and artificial intelligence.

In this paper, we restrict the subject of investigation to jokes, one of the most common humor types
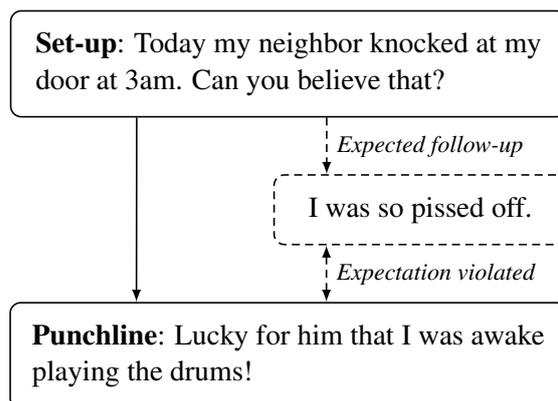

Figure 1: A joke example consisting of a set-up and a punchline. A violation can be observed between the punchline and the expectation.

in text form. As shown in Figure 1, these jokes usually consist of a *set-up* and a *punchline*. The set-up creates a situation that introduces the hearer into the story framework, and the punchline concludes the joke in a succinct way, intended to make the hearer laugh. Perhaps the most suitable humor theory for explaining such humor phenomenon is the *incongruity theory*, which states that the cause of laughter is the perception of something incongruous (the punchline) that violates the hearer's expectation (the set-up).

Based on the incongruity theory, we propose two features for humor recognition, by calculating the degree of incongruity between the set-up and the punchline. Recently popular pre-trained language models enable us to study such relationship based on large-scale corpora. Specifically, we fed the set-up along with the punchline into the GPT-2 language model (Radford et al., 2019), and obtained the surprisal and uncertainty values of the joke, indicating how surprising it is for the model to generate the punchline, and the uncertainty while generating it. We conducted experiments on a manually labeled humor dataset, and the results showed that

these two features could better distinguish jokes from non-jokes, compared with existing baselines. Our work made an attempt to bridge humor theories and humor recognition by applying large-scale pre-trained language models, and we hope it could inspire future research in computational humor.

2 Related Work

Humor Data Mihalcea and Strapparava (2005) created a one-liner dataset with humorous examples extracted from webpages with humor theme and non-humorous examples from Reuters titles, British National Corpus (BNC) sentences, and English Proverbs. Yang et al. (2015) scraped puns from the Pun of the Day website[1] and negative examples from various news websites. There is also work on the curation of non-English humor datasets (Zhang et al., 2019; Blinov et al., 2019). Hasan et al. (2019) developed UR-FUNNY, a multimodal humor dataset that involves text, audio and video information extracted from TED talks.

Humor Recognition Most of the existing work on humor recognition in text focuses on one-liners, one type of jokes that delivers the laughter in a single line. The methodologies typically fall into two categories: feature engineering and deep learning. Mihalcea and Strapparava (2005) designed three human-centric features (alliteration, antonymy and synonym) for recognizing humor in the curated one-liner dataset. Mihalcea et al. (2010) approached the problem by calculating the semantic relatedness between the set-up and the punchline (they evaluated 150 one-liners by manually splitting them into "set-up" and "punchline"). Shahaf et al. (2015) investigated funny captions for cartoons and proposed several features including perplexity to distinguish between funny and less funny captions. Morales and Zhai (2017) proposed a probabilistic model and leveraged background text sources (such as Wikipedia) to identify humorous Yelp reviews. Liu et al. (2018) proposed to model sentiment association between elementary discourse units and designed features based on discourse relations. Cattle and Ma (2018) explored the usage of word associations as a semantic relatedness feature in a binary humor classification task. With neural networks being popular in recent years, some deep learning structures have been developed for the recognition of humor in text. Chen and Lee (2017) and

---

[1] http://www.punoftheday.com/

Chen and Soo (2018) adopted convolutional neural networks, while Weller and Seppi (2019) used a Transformer architecture to do the classification task. Fan et al. (2020) incorporated extra phonetic and semantic (ambiguity) information into the deep learning framework. In addition to these methodological papers, there are also some tasks dedicated to computational humor in recent years. SemEval 2020 Task 7 (Hossain et al., 2020) aims at assessing humor in edited news headlines. SemEval 2021 Task 7 (Meaney et al., 2021) involves predicting the humor rating of the given text, and if the rating is controversial or not. In this task, Xie et al. (2021) adopted the DeBERTa architecture (He et al., 2020) with disentangled attention mechanism to predict the humor labels.

Although the work of Mihalcea et al. (2010) is the closest to ours, we are the first to bridge the incongruity theory of humor and large-scale pre-trained language models. Other work (Bertero and Fung, 2016) has attempted to predict punchlines in conversations extracted from TV series, but their subject of investigation should be inherently different from ours—punchlines in conversations largely depend on the preceding utterances, while jokes are much more succinct and self-contained.

3 Humor Theories

The attempts to explain humor date back to the age of ancient Greece, where philosophers like Plato and Aristotle regarded the enjoyment of comedy as a form of scorn, and held critical opinions towards laughter. These philosophical comments on humor were summarized as the *superiority theory*, which states that laughter expresses a feeling of superiority over other people's misfortunes or shortcomings. Starting from the 18th century, two other humor theories began to challenge the dominance of the superiority theory: the *relief theory* and the *incongruity theory*. The relief theory argues that laughter serves to facilitate the relief of pressure for the nervous system (Morreall, 2020). This explains why laughter is caused when people recognize taboo subjects—one typical example is the wide usage of sexual terms in jokes. The incongruity theory, supported by Kant (1790), Schopenhauer (1883), and many later philosophers and psychologists, states that laughter comes from the perception of something incongruous that violates the expectations. This view of humor fits well the types of jokes commonly found in stand-up comedies,

where the set-up establishes an expectation, and then the punchline violates it. As an expansion of the incongruity theory, Raskin (1979) proposed the Semantic Script-based Theory of Humor (SSTH) by applying the semantic script theory. It posits that, in order to produce verbal humor, two requirements should be fulfilled: (1) The text is compatible with two different scripts; (2) The two scripts with which the text is compatible are opposite.

4 Methodology

The incongruity theory attributes humor to the violation of expectation. This means the punchline delivers the incongruity that turns over the expectation established by the set-up, making it possible to interpret the set-up in a completely different way. With neural networks blooming in recent years, pretrained language models make it possible to study such relationship between the set-up and the punchline based on large-scale corpora. Given the set-up, language models are capable of writing expected continuations, enabling us to measure the degree of incongruity, by comparing the actual punchline with what the language model is likely to generate.

In this paper, we leverage the GPT-2 language model (Radford et al., 2019), a Transformer-based architecture trained on the WebText dataset. We chose GPT-2 because: (1) GPT-2 is already pretrained on massive data and publicly available online, which spares us the training process; (2) it is domain independent, thus suitable for modeling various styles of English text. Our goal is to model the set-up and the punchline as a whole piece of text using GPT-2, and analyze the probability of generating the punchline given the set-up. In the following text, we denote the set-up as x, and the punchline as y. Basically, we are interested in two quantities regarding the probability distribution $p(y|x)$: uncertainty and surprisal, which are elaborated in the next two sections.

4.1 Uncertainty

The first question we are interested in is: given the set-up, how uncertain it is for the language model to continue? This question is related to SSTH, which states that, for a piece of text to be humorous, it should be compatible with two different scripts. To put it under the framework of set-up and punchline, this means the set-up could have multiple ways of interpretation, according to the following punchline. Thus, one would expect a higher uncertainty
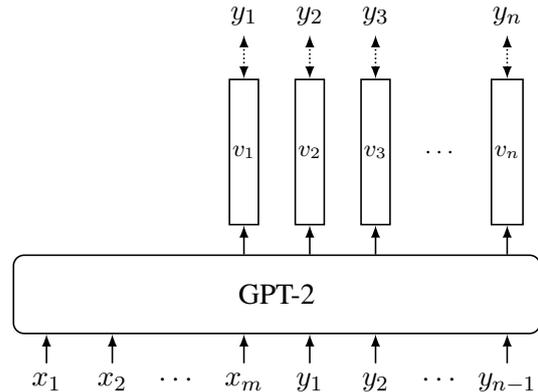

Figure 2: The set-up x and the punchline y are concatenated and fed into GPT-2 for predicting the next token. v_i's are probability distributions on the vocabulary.

value when the language model tries to continue the set-up and generate the punchline.

We propose to calculate the averaged entropy of the probability distributions at all token positions of the punchline, to represent the degree of uncertainty. As shown in Figure 2, the set-up x and the punchline y are concatenated and then fed into GPT-2 to predict the next token. While predicting the tokens of y, GPT-2 produces a probability distribution v_i over the vocabulary. The averaged entropy is then defined as

$$U(x,y) = -\frac{1}{|y|} \sum_{i=1}^{n} \sum_{w \in V} v_i^w \log v_i^w, \quad (1)$$

where V is the vocabulary.

4.2 Surprisal

The second question we would like to address is: how surprising it is when the language model actually generates the punchline? As the incongruity theory states, laughter is caused when something incongruous is observed and it violates the previously established expectation. Therefore, we expect the probability of the language model generating the actual punchline to be relatively low, which indicates the surprisal value should be high. Formally, the surprisal is defined as

$$S(x,y) = -\frac{1}{|y|} \log p(y|x)$$
$$= -\frac{1}{|y|} \sum_{i=1}^{n} \log v_i^{y_i}. \quad (2)$$

5 Experiments

We evaluated and compared the proposed features with several baselines by conducting experiments

in two settings: predicting using individual features, and combining the features with a content-based text classifier.

5.1 Baselines

Similar to our approach of analyzing the relationship between the set-up and the punchline, Mihalcea et al. (2010) proposed to calculate the semantic relatedness between the set-up and the punchline. The intuition is that the punchline (which delivers the surprise) will have a minimum relatedness to the set-up. For our experiments, we chose two relatedness metrics that perform the best in their paper as our baselines, plus another similarity metric based on shortest paths in WordNet (Miller, 1995):

- **Leacock & Chodorow similarity** (Leacock and Chodorow, 1998), defined as

$$\text{Sim}_{lch} = -\log \frac{length}{2 * D},\tag{3}$$

  where *length* is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of WordNet.

- **Wu & Palmer similarity** (Wu and Palmer, 1994) calculates similarity by considering the depths of the two synsets in WordNet, along with the depth of their *LCS* (Least Common Subsumer), which is defined as

$$\text{Sim}_{wup} = \frac{2 * \text{depth}(LCS)}{\text{depth}(C_1) + \text{depth}(C_2)},\tag{4}$$

  where C_1 and C_2 denote synset 1 and synset 2 respectively.

- **Path similarity** (Rada et al., 1989) is also based on the length of the shortest path between two concepts in WordNet, which is defined as

$$\text{Sim}_{path} = \frac{1}{1 + length}.\tag{5}$$

In addition to the metrics mentioned above, we also consider the following two baselines related to the phonetic and semantic styles of the input text:

- **Alliteration**. The alliteration value is computed as the total number of alliteration chains and rhyme chains found in the input text (Mihalcea and Strapparava, 2005).

- **Ambiguity**. Semantic ambiguity is found to be a crucial part of humor (Miller and Gurevych, 2015). We follow the work of Liu et al. (2018) to compute the ambiguity value:

$$\log \prod_{w \in s} \text{num\_of\_senses}(w),\tag{6}$$

  where w is a word in the input text s.

5.2 Dataset

We took the dataset from SemEval 2021 Task 7.[2] The released training set contains 8,000 manually labeled examples in total, with 4,932 being positive, and 3,068 negative. To adapt the dataset for our purpose, we only considered positive examples with exactly two sentences, and negative examples with at least two sentences. For positive examples (jokes), the first sentence was treated as the set-up and the second the punchline. For negative examples (non-jokes), consecutive two sentences were treated as the set-up and the punchline, respectively.[3] After splitting, we cleaned the data with the following rules: (1) We restricted the length of set-ups and punchlines to be under 20 (by counting the number of tokens); (2) We only kept punchlines whose percentage of alphabetical letters is greater than or equal to 75%; (3) We discarded punchlines that do not begin with an alphabetical letter. As a result, we obtained 3,341 examples in total, consisting of 1,815 jokes and 1,526 non-jokes. To further balance the data, we randomly selected 1,526 jokes, and thus the final dataset contains 3,052 labeled examples in total. For the following experiments, we used 10-fold cross validation, and the averaged scores are reported.

5.3 Predicting Using Individual Features

To test the effectiveness of our features in distinguishing jokes from non-jokes, we built an SVM classifier (parameters can be found in Appendix A) for each individual feature (uncertainty and surprisal, plus the baselines). The resulted scores are reported in Table 1. Compared with the baselines, both of our features (uncertainty and surprisal) achieved higher scores for all the four metrics. In addition, we also tested the performance of uncertainty combined with surprisal (last row

36

	P	R	F1	Acc
Random	0.4973	0.4973	0.4958	0.4959
Sim_{lch}	0.5291	0.5179	0.4680	0.5177
Sim_{wup}	0.5289	0.5217	0.4919	0.5190
Sim_{path}	0.5435	0.5298	0.4903	0.5291
Alliteration	0.5353	0.5349	0.5343	0.5354
Ambiguity	0.5461	0.5365	0.5127	0.5337
Uncertainty	0.5840	0.5738	0.5593	0.5741
Surprisal	0.5617	0.5565	0.5455	0.5570
U+S	**0.5953**	**0.5834**	**0.5695**	**0.5832**

Table 1: Performance of individual features. Last row (U+S) is the combination of uncertainty and surprisal. P: Precision, R: Recall, F1: F1-score, Acc: Accuracy. P, R, and F1 are macro-averaged, and the scores are reported on 10-fold cross validation.

	P	R	F1	Acc
GloVe	0.8233	0.8232	0.8229	0.8234
GloVe+Sim_{lch}	0.8255	0.8251	0.8247	0.8250
GloVe+Sim_{wup}	0.8264	0.8260	0.8254	0.8257
GloVe+Sim_{path}	0.8252	0.8244	0.8239	0.8244
GloVe+Alliter.	0.8299	0.8292	0.8291	0.8297
GloVe+Amb.	0.8211	0.8203	0.8198	0.8201
GloVe+U	0.8355	0.8359	0.8353	0.8359
GloVe+S	0.8331	0.8326	0.8321	0.8326
GloVe+U+S	**0.8368**	**0.8368**	**0.8363**	**0.8365**

Table 2: Performance of the features when combined with a content-based classifier. U denotes uncertainty and S denotes surprisal. P: Precision, R: Recall, F1: F1-score, Acc: Accuracy. P, R, and F1 are macro-averaged, and the scores are reported on 10-fold cross validation.

of the table), and the resulting classifier shows a further increase in the performance. This suggests that, by jointly considering uncertainty and surprisal of the set-up and the punchline, we are better at recognizing jokes.

5.4 Boosting a Content-Based Classifier

Now that we have shown the advantage of our features when used individually in prediction, we would like to validate their effectiveness when combined with the commonly used word embeddings. Thus, we evaluated our features as well as the baselines under the framework of a content-based classifier. The idea is to see if the features could further boost the performance of existing text classifiers. To create a starting point, we encoded each set-up and punchline into vector representations by aggregating the GloVe (Pennington et al., 2014) embeddings of the tokens (sum up and then normalize by the length). We used the GloVe embeddings
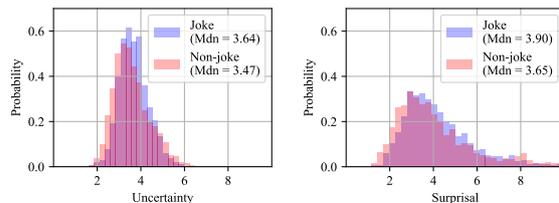

Figure 3: Histograms of uncertainty (left) and surprisal (right), plotted separately for jokes and non-jokes. Mdn stands for Median.

with dimension 50, and then concatenated the set-up vector and the punchline vector, to represent the whole piece of text as a vector of dimension 100. For each of the features (uncertainty and surprisal, plus the baselines), we appended it to the GloVe vector, and built an SVM classifier to do the prediction. Scores are reported in Table 2. As we can see, compared with the baselines, our features produce larger increases in the performance of the content-based classifier, and similar to what we have observed in Table 1, jointly considering uncertainty and surprisal gives further increase in the performance.

6 Visualizing Uncertainty and Surprisal

To get a straightforward vision of the uncertainty and surprisal values for jokes versus non-jokes, we plot their histograms in Figure 3 (for all 3,052 labeled examples). It can be observed that, for both uncertainty and surprisal, jokes tend to have higher values than non-jokes, which is consistent with our expectations in Section 4.

7 Conclusion

This paper makes an attempt in establishing a connection between the humor theories and the nowadays popular pre-trained language models. We proposed two features according to the incongruity theory of humor: uncertainty and surprisal. We conducted experiments on a humor dataset, and the results suggest that our approach has an advantage in humor recognition over the baselines. The proposed features can also provide insight for the task of two-line joke generation—when designing the text generation algorithm, one could exert extra constraints so that the set-up is chosen to be compatible with multiple possible interpretations, and the punchline should be surprising in a way that violates the most obvious interpretation. We hope our work could inspire future research in the community of computational humor.

References

Dario Bertero and Pascale Fung. 2016. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of NAACL-HLT 2016*, pages 130–135.

Vladislav Blinov, Valeria Bolotova-Baranova, and Pavel Braslavski. 2019. Large dataset and language model fun-tuning for humor recognition. In *Proceedings of ACL 2019*, pages 4027–4032.

Andrew Cattle and Xiaojuan Ma. 2018. Recognizing humour using word associations and humour anchor extraction. In *Proceedings of COLING 2018*, pages 1849–1858.

Lei Chen and Chong Min Lee. 2017. Convolutional neural network for humor recognition. *CoRR*, abs/1702.02584.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of NAACL-HLT 2018, Volume 2 (Short Papers)*, pages 113–117.

Xiaochao Fan, Hongfei Lin, Liang Yang, Yufeng Diao, Chen Shen, Yonghe Chu, and Tongxuan Zhang. 2020. Phonetics and ambiguity comprehension gated attention network for humor recognition. *Complex.*, 2020:2509018:1–2509018:9.

Md. Kamrul Hasan, Wasifur Rahman, AmirAli Bagher Zadeh, Jianyuan Zhong, Md. Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed (Ehsan) Hoque. 2019. UR-FUNNY: A multimodal language dataset for understanding humor. In *Proceedings of EMNLP-IJCNLP 2019*, pages 2046–2056.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Nabil Hossain, John Krumm, Michael Gamon, and Henry A. Kautz. 2020. SemEval-2020 Task 7: Assessing humor in edited news headlines. In *Proceedings of SemEval@COLING 2020*, pages 746–758.

Immanuel Kant. 1790. Critique of judgment, ed. and trans. *WS Pluhar, Indianapolis: Hackett*.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and WordNet sense similarity for word sense identification. In *WordNet, An Electronic Lexical Database*. The MIT Press.

Lizhen Liu, Donghai Zhang, and Wei Song. 2018. Modeling sentiment association in discourse for humor recognition. In *Proceedings of ACL 2018, Volume 2 (Short Papers)*, pages 586–591.

J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 Task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of SemEval@ACL 2021*.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of HLT/EMNLP 2005*, pages 531–538.

Rada Mihalcea, Carlo Strapparava, and Stephen G. Pulman. 2010. Computational models for incongruity detection in humour. In *Proceedings of CICLing 2010*, volume 6008 of *Lecture Notes in Computer Science*, pages 364–374.

George A. Miller. 1995. Wordnet: A lexical database for English. *Commun. ACM*, 38(11):39–41.

Tristan Miller and Iryna Gurevych. 2015. Automatic disambiguation of English puns. In *Proceedings of ACL 2015*, pages 719–729.

Alex Morales and Chengxiang Zhai. 2017. Identifying humor in reviews using background text sources. In *Proceedings of EMNLP 2017*, pages 492–501.

John Morreall. 2020. Philosophy of Humor. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, fall 2020 edition. Metaphysics Research Lab, Stanford University.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543.

Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Trans. Syst. Man Cybern.*, 19(1):17–30.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Raskin. 1979. Semantic mechanisms of humor. In *Annual Meeting of the Berkeley Linguistics Society*, volume 5, pages 325–335.

Arthur Schopenhauer. 1883. The world as will and idea (vols. i, ii, & iii). *Haldane, RB, & Kemp, J.(3 Vols.). London: Kegan Paul, Trench, Trubner*, 6.

Dafna Shahaf, Eric Horvitz, and Robert Mankoff. 2015. Inside jokes: Identifying humorous cartoon captions. In *Proceedings of SIGKDD 2015*, pages 1065–1074.

Orion Weller and Kevin D. Seppi. 2019. Humor detection: A transformer gets the last laugh. In *Proceedings of EMNLP-IJCNLP 2019*, pages 3619–3623.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of ACL 1994*, page 133–138.

Yubo Xie, Junze Li, and Pearl Pu. 2021. HumorHunter at SemEval-2021 Task 7: Humor and offense recognition with disentangled attention. In *Proceedings of SemEval@ACL 2021*.

	Running Time
Sim_{lch}	1.76 sec
Sim_{wup}	1.71 sec
Sim_{path}	1.71 sec
Alliteration	1.70 sec
Ambiguity	2.94 sec
Uncertainty	2.12 sec
Surprisal	2.49 sec
Uncertainty + Surprisal	2.26 sec

Table 3: Running time of the SVM classifiers trained on individual features.

	Running Time
GloVe	7.54 sec
GloVe + Sim_{lch}	14.85 sec
GloVe + Sim_{wup}	15.90 sec
GloVe + Sim_{path}	13.76 sec
GloVe + Alliteration	15.41 sec
GloVe + Ambiguity	14.28 sec
GloVe + Uncertainty	14.70 sec
GloVe + Surprisal	13.84 sec
GloVe + U + S	19.27 sec

Table 4: Running time of the content-based SVM classifiers combined with individual features. U denotes uncertainty and S denotes surprisal.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard H. Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of EMNLP 2015*, pages 2367–2376.

Dongyu Zhang, Heting Zhang, Xikai Liu, Hongfei Lin, and Feng Xia. 2019. Telling the whole story: A manually annotated chinese dataset for the analysis of humor in jokes. In *Proceedings of EMNLP-IJCNLP 2019*, pages 6401–6406.

A Model Parameters

For the SVM classifier, we set the regularization parameter $C = 1.0$, and used the RBF kernel with the kernel coefficient $\gamma = 1/n_{\text{features}}$. All models were trained and evaluated on a machine with Intel Core i7-6700K CPU, Nvidia GeForce GTX 1080 GPU, and 16GB RAM. The running time of each method is listed in Table 3 and Table 4.