CSD: A Chinese Dataset for Subtext Problem

Anonymous ACL submission

Abstract

Subtext is a kind of deep semantics which can be acquired after one or more rounds of expression transformation. As a popular way of expressing one's intentions, it is well worth studying. In this paper, we propose two subtext-related tasks which are termed "subtext recognition" and "subtext recovery" and make a clear definition for their purposes. Moreover, we build a Chinese dataset whose source data comes from popular social media (e.g. Weibo, Netease Music, Zhihu, and Bilibili) and propose a new evaluation metric termed "Two-stages Annotation Evaluation" (TAE) for the validation of a multi-turn annotation process.

1 Introduction

Subtext is a kind of deep semantics for expressing emotions, describing opinions and conveying intention, which is widely used in text and conversations. However, the subtext can not be directly obtained from the text sequence, which makes it difficult to be analyzed by machine learning methods. As far as we know, in the field of natural language processing (NLP), the research on subtext has never been mentioned. In this paper, we put forward the concept and metrics of subtext analysis and divide it into two tasks, i.e., subtext recognition and subtext recovery.

The subtext is widespread in English and Chinese where the two languages meet high level agreement on the definitions, which can be summarized as "implicit meaning of a text, often a literary one, a speech, or a dialogue". The "implicit meaning" is a kind of deep semantics obtained after one or more transformations. We define some notations to represent this process, s is a sequence containing subtext, c is a context sequence or background of s, $f(\cdot)$ is the function of extracting the sequence meaning and c_o is the common knowledge. A sequence contains subtext when $f(s|c_o) \neq f(c \oplus s|c_o)$, where $\oplus$ denotes the process of information fusion. As the common knowledge is always encoded into the embedding of text, we abbreviate the formula as $f(s) \neq f(c \oplus s)$. c represents the previous and latter sentences of the target, or the description of background information of the target. In advance, we can obtain the original meaning of a sequence by the following steps. 1.If there are rhetoric words, replace them with plain words. For example, replace metaphorical words with the corresponding entities, and replace satirical words with negative forms; 2. Analyse the result of the first step, and infer the deep hidden meaning, which is related to the context and background knowledge of other complex text information.

We show some examples containing subtext to facilitate understanding in Table 1, where the column 'comment' is the target sequence to analyze, and the column 'context' is the context of comment. The first example in Table 1 means that I will overcome difficulties and become your boyfriend, not that I want to cross a wall literally. There is background information that this comment is about a love song from NetEase Cloud music. Therefore, we take the background as c. The meaning of s on with c is "I love you no matter what difficulties I will meet". According to the definition, $f(s) \neq f(c \oplus s)$ so that the target sequence s contains subtext. The original meaning of s can be obtained by the following process: firstly, replace the metaphorical words "墙(the wall)" and "跨过(cross)" with "difficulties" and "overcome", respectively; secondly, infer according to the context. We get that the speaker wants to overcome difficulties to go into another girl's heart, which can be concluded as "I want to be your girlfriend/boyfriend, and I will overcome difficulties". This example contains metaphors. However, metaphor only makes us get that "wall" means "difficulty", but subtext analysis makes us get that "I love you". To show that subtext does not appear with metaphor, we

Table 1: Annotation samples, where "subt", "sarc", "meta", "exag", "homo", "emot", "sent", "other" stand for "subtext", "sarcasm", "metaphor", "exaggeration", "homophonic", "emotion", "sentiment" and other kinds of "rhetoric" respectively.

No.	comment	context	subt	sarc	meta	exag	homo	emot	sent	other
1	你的心里有一道墙，我要跨过这道墙(If there is a wall in your heart, I want to cross it)	来自网易云的情歌。 (A love song from NetEase Cloud music.)	1	-1	1	-1	-1	anticipation	0	-1
2	你隔岸观火却不救我 (You watched the fire from the other side but didn't save me)	来自网易云的情歌。 (A love song from NetEase Cloud music.)	1	0	1	-1	-1	sad	-1	-1
3	好家伙，起码9年 (On my god! At least nine years)	1个月一期，一共一百期，追个几年没问题了[doge] (The video series is updated once a month, and there are 100 items in total. It's no problem to chase it for a few years [doge])	1	1	-1	-1	-1	None	0	-1

take the third case in Table 1 for example. s literally means that the time is at least nine years. The meaning of s under the context is that the video is updated so slowly. According to the definition, $f(s) \neq f(c \oplus s)$ so that the target sequence s contains subtext. Besides, "I burn you" is another example in English, who contains subtext without metaphor. It means that we win the debate.

This paper introduces an enlightening research to determine whether a sentence contains subtext. We define the process of learning a judge function J for whether $f(s)$ equals to $f(c \oplus s)$ as *subtext recognition*, which is a sub-field of text classification. The research of the transformation process for obtaining the original meaning of subtext is defined as *subtext recovery*.

In this paper, a Chinese corpus for subtext recognition is constructed. This dataset provides both coarse-grained and fine-grained labeling for each Chinese sequence, where the coarse-grained labeling contains *subtext* (0) or *non-subtext* (1) and the fine-grained labeling can be used to recognize what kinds of *subtext* it is.

Contribution: As far as we know, we are the first to analyze whether a sentence contains subtext. Such analysis empowers machines to know what people really mean, which can make machine translation and sentiment analysis more accurate. Our contribution in this paper can be summarized as follows:

- We put forward text subtext analysis in the field of NLP and divide it into two tasks which can be solved by the existing NLP models.
- We build a **C**hinese **s**ubtext **d**ataset (CSD-Dataset) from popular social media, and evaluate its quality. Moreover, CSD-Dataset can be used in many tasks including emotion analysis, subtext analysis, sarcasm analysis, metaphor analysis, homophonic analysis and exaggeration analysis.
- We evaluate the reliablility of CSD-Dataset by three different methods and make a detailed statistical analysis on it.

2 Related Work

2.1 Dataset Construction

(Kant et al., 2018) constructs their emotional dataset by proposing a multidimensional model, classifying the feelings into eight classes including *anger*, *fear*, *disgust*, *trust*, *joy*, *surprise*, *anticipation*, and *sad*, which meet high agreement with the setting in (Plutchik, 1984). (Lin and Hsieh, 2016) uses a crowdsourcing method to build sarcasm corpus. In (Nakov et al., 2013), they annotate the sentence with three classes including positive, negative and neutral or objective. (Rosenthal et al., 2017) labels each tweet as positive, negative, neutral or junk. (Öhman et al., 2020) constructs a dataset for sentiment analysis and emotion analysis, in which the authors evaluate the dataset by constructing a classification model based on Support Vector Machine or BERT (Devlin et al., 2019). We construct our data-set according to the works mentioned above.

2.2 Related Task

Metaphor Detection: The main purpose of metaphor detection (Rei et al., 2017) is to judge whether a sentence contains metaphorical objects. In advance, token-level metaphor analysis (Stowe and Palmer, 2018; Mosolova et al., 2018; Mao et al., 2019) attempts to recognize the position of ontology and metaphorical objects in a sentence, similar to the first step of subtext analysis. However, the goals of them are different. The goal of subtext

analysis is judging whether a sentence contains implicit meaning. Different from metaphor detection, subtext analysis will find the position of subsequence of text which conveys the implicit meaning. Take the first example in Table 1 for example, metaphor analysis identifies"wall" as the figurative expression of "obstacles". However subtext analysis identifies "I love you". Moreover, from the examples in the introduction, it can be easily found that there is no relation between inclusion and being included for subtext and metaphor.

Sentiment Analysis: The purpose of sentiment analysis is to analyze the attitudes, sentiments, emotions and so on of people (Zhang et al., 2018; Liu, 2012). In general, the sentiment analysis is either to judge whether the attitudes are positive or negative, or to judge whether the emotions are happiness or sadness or other kinds of feelings.

MultiWords Expression and Idiomatic Analysis: Multiwords Expression (MWEs) is a basic semantic unit, which meets high agreement of conventionality accordding to (Calzolari et al., 2002), which can be used as a feature of subtext analysis. For example, "隔岸观火 (You do not help me while I am struggling.)" is a multiwords expression in example 2. Idiomatic analysis (Levorato et al., 2004) analyses a fix expressions, of which the meaning is consistent in different context. But it's not suitable for subtext. Although the words and orders are the same, different contexts will determine whether there is a subtext. Besides, MWEs and idiomatic analysis denote the language phenomenon that $f(s) = f(c \oplus s)$, which shows that our definition is reasonable inversely.

3 Chinese Subtext Dataset

3.1 Data Collection

To collect as much data as possible, we grabbed the comment data from the hot lists of four major websites (Weibo, NetEase Music, BiliBili and Zhihu), in which the quality of comments is higher than other lists. In order to analyze the comment with multiple sub-comments, we retained the structure information of the source comment, including comment, comment ID, context, context ID, and the source of comment. Finally, we collected about 70,000 comments.

3.2 Annotation Processing

Anonymity: To protect the personal identity information of users, we remove the annotation ID, parent ID and the nickname of users from a piece of data. But we do not remove all names in comments which are popular in our daily life, such as the name of an idol.

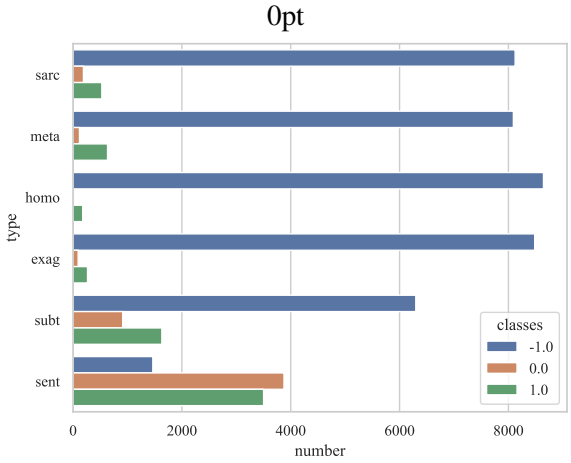


Figure 1: The ratio of different labeling information. The horizontal axis represents numbers, and the vertical axis represents different labels.

Annotation methods: To prevent subjective influence, each comment was labeled by three people independently and then the three different labels were checked by other people. We call it **T**wo-**S**tages **L**abeling Process (TSL). In order to avoid the diversity of expression, we have defined all category labels, and only one of them can be selected as the category results for annotations. In the first phase of TSL, three people are asked to label independently. In the second phase of TSL, the fourth person annotates the text according to the three labeling results. There are several situations in the second phase: 1. all the labels are the same, then we adopt them; 2. all the labels are different, then we delete this comment or relabel this comment; 3. Part of labels are different, then we re-annotate it; 4. if one data is labeled twice and it still specious, we delete it.

Category Label Some of the annotation samples are displayed in Table 1. Subtext is often accompanied with some rhetorical words, so we label the most commonly used rhetorical information, which can be used to identify and analyze subtext. Therefore, we annotate a comment with eight kinds of information: *subtext*, *sarcasm*, *metaphor*, *exaggeration*, *homophonic*, *sentiment*, *emotion*, and *other*, where *other* is the unified category of other rhetorical methods that appear less than 50 times. In details, subtext, sarcasm, metaphor, exaggeration, homophonic and sentiment are marked with three

tags: Tag (1) means that the sentence contains the corresponding information; Tag (-1) means that the sentence does not contain the corresponding information; Tag (0) means unsure. We label 8 kinds of emotional information as (Kant et al., 2018; Plutchik, 1984) did. Moreover, we also add *None* to indicate that there is no obvious emotional expression in the text. The sentiments are annotated like (Nakov et al., 2013; Rosenthal et al., 2017). Eventually, we get 8843 annotated comments after removing useless data.

Table 2: The score of different evaluation methods.

type	sarc	meta	subt	exag	homo	sent	other
Kappa	0.60	0.60	0.60	0.71	0.61	0.73	0.26
TAE	0.81	0.56	0.50	0.88	0.95	0.64	0.93
$\mathbf{F}_{1,SVM}$	0.51	0.50	0.47	0.50	0.53	0.54	0.49

3.3 Quality Evaluation

In order to ensure the quality of labeling, TSL is adopted in this paper. To evaluate the reliability, we uses Kappa score as (Ghanem et al., 2019; Khodak et al., 2018; Webster et al., 2018) did. When using Kappa score, we treat the four annotation results equally. Besides, to follow (Öhman et al., 2020), we use F_1 score whose inputs are predictions generated by Support Vector Machine (SVM) and ground truth annotated by the fourth annotator, as one of evaluation metrics. The implementation of SVM and the computation of F_1 score are supported by scikit-learn library (Pedregosa et al., 2011). Moreover, we train the SVM classifier by 5-fold-cross-validation. Considering the shortcoming of Kappa score (Artstein and Poesio, 2008; Sim and Wright, 2005): if the data is extremely imbalanced, the Kappa score will be low, even when the annotations meet high agreement, this paper introduces an evaluation metric which is termed the **T**wo-Stages **A**nnotation **E**valuation (TAE). To avoid the influence of data imbalance, TAE evaluates the dataset by a sequence-wise score and calculates the average of the whole sequence. We put the details of TAE in Appendix A.

3.4 Dataset Analysis

Figure 1 displays the ratio of different classes for different labeling information. It is obvious that the distribution of different classes is extremely imbalanced, which will cause a problem of a high agreement but a low score in Kappa (Sim and Wright, 2005; Feinstein and Cicchetti, 1990). Therefore, we combine the evaluation results of TAE score , Kappa score and F_1 score and come to the conclusion that our corpus is reliable. The evaluation results of three different evaluation metrics are shown in Table 2. All of them show that our dataset is reliable. The range of F_1 score is from 0.47 to 0.54, where subtext gets the lowest score, sentiment gets the highest score. The score of *sarcasm*, *exaggeration*, *homophonic* and *other* are high in TAE, but is low in Kappa for the degrees of which the corresponding data imbalance are more obvious. The scores of the evaluation metrics for subtext show that it is tougher to annotate than other information. Besides, we analyze the overlapping situation of instances in one data type and another. The overlapping results are displayed in Figure 2, where the metaphor as the closest task to subtext only shows half overlapping ratios. Therefore, the subtext is relatively independent from other semantic features. Moreover, we compute the Spearman coefficient between different classes as shown in Figure 3 where subtext is more related to sarcasm (0.53) and metaphor (0.55) than others.

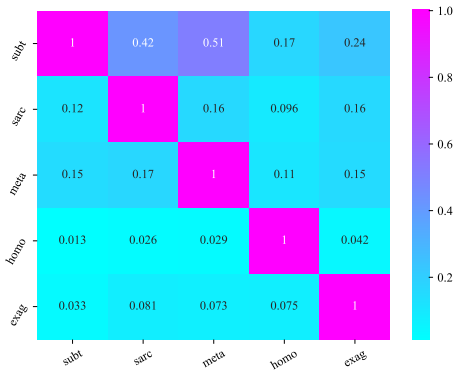


Figure 2: The ratios of overlap between type1 and type2.

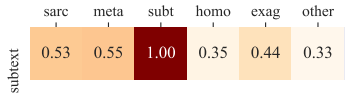


Figure 3: The Spearman coefficients.

4 Conclusion

In this paper, we put forward the new problems which are called "subtext recognition" and "subtext recovery". We collect data from the popular Chinese social media, annotate them for many tasks including subtext analysis, sarcasm analysis and so on, and use three methods to evaluate the reliability. We also propose a new evaluation metric, TAE, to avoid the impact of data imbalance.

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Nicoletta Calzolari, Charles J. Fillmore, Ralph Grishman, Nancy Ide, Alessandro Lenci, Catherine Macleod, and Antonio Zampolli. 2002. Towards best practice for multiword expressions in computational lexicons. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain*. European Language Resources Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Alvan R Feinstein and Domenic V Cicchetti. 1990. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. 2019. Idat at fire2019: Overview of the track on irony detection in arabic tweets. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 10–13.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Neel Kant, Raul Puri, Nikolai Yakovenko, and Bryan Catanzaro. 2018. Practical text classification with large pre-trained language models. *arXiv preprint*, arXiv:1812.01207.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Maria Chiara Levorato, Barbara Nesi, and Cristina Cacciari. 2004. Reading comprehension and understanding idiomatic expressions: A developmental study. *Brain and Language*, 91(3):303–314.

Shih-Kai Lin and Shu-Kai Hsieh. 2016. Sarcasm detection in chinese using a crowdsourced corpus. In *Proceedings of the Conference on Computational Linguistics and Speech Processing*, pages 299–310.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Rui Mao, Chenghua Lin, and Frank Guerin. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3888–3898.

Anna Mosolova, Ivan Bondarenko, and Vadim Fomin. 2018. Conditional random fields for metaphor detection. In *Proceedings of the Workshop on Figurative Language Processing*, pages 121–123.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 312–320.

Emily Öhman, Marc Pàmies, Kaisla Kajava, and Jörg Tiedemann. 2020. XED: A multilingual dataset for sentiment analysis and emotion detection. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6542–6552. International Committee on Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1532–1543.

Robert Plutchik. 1984. Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219.

Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. Grasping the finer point: A supervised similarity network for metaphor detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1537–1546.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 502–518.

Julius Sim and Chris C Wright. 2005. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3):257–268.

Kevin Stowe and Martha Palmer. 2018. Leveraging syntactic constructions for metaphor identification. In *Proceedings of the Workshop on Figurative Language Processing*, pages 17–26.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced

corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(4).

A Appendix

A.1 The Details of TAE

TAE Score: To compute TAE score, we first define the two values for single annotation records.

- Agreement: It is defined as the ratios of labeling results in the first round with agreement to the second round and the number of the possible labels. Let $L_1 = \{l_{11}, l_{12}, \cdots, l_{1n}\}$ be the labeling results of the first round annotation, l_2 be the labeling result of the second round annotation. The agreement for the i-th records is computed as:

$$\text{agr}_i = \frac{|\{l_{1j} | l_{1j} == l_2; j=1,2,\cdots,n\}|}{n}. \quad (1)$$

- Randomness: It is defined as the ratios of the number of all the label types in the first round without the the second round agreement and the number of all the possible label types. Let ls(s) be the function of turning a list into a set, and s be a list. Let $\text{Li}_1 = [l_{11}, l_{12}, \cdots, l_{1n}]$ be the tabular form of L_1, and Li_2 to be tabular form of $L_2 = \{l_2\}$. The randomness for the i-th records is computed as:

$$\text{rad}_i = \frac{\text{ls}(\text{Li}_1) \setminus \text{ls}(\text{Li}_2)}{\text{ls}(\text{Li}_1) \cup \text{ls}(\text{Li}_2)}. \quad (2)$$

To be a validation metric, TAE should satisfy the following properties:

- Monotony. TAE score should be monotonically increasing with respect to the agreement. TAE score should be monotonically decreasing with respect to the randomness as well.
- Boundness. TAE score should be robust about randomness and consistency so that we can measure whether a data set is reliable.
- Independence. TAE score should be independent of the ratio of positive samples and negative samples, which is the main shortcomings of Kappa score.

Consequently, we define TAE score as follows:

$$\text{TAE} = \frac{\exp(agr - rad) - 1/e}{e - 1/e} \quad (3)$$

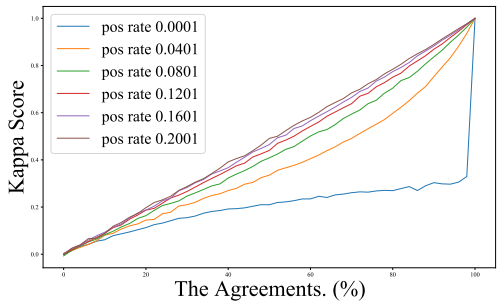
$$\text{agr} = \frac{\sum_{i=1}^{n} agr_i}{n} \quad (4)$$

$$\text{rad} = \frac{\sum_{i=1}^{n} rad_i}{n}. \quad (5)$$

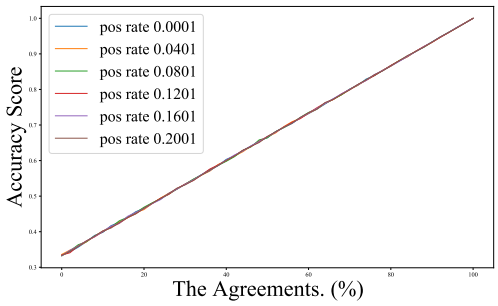
To illustrate TAE score is valid, we make some simulation experiments. We execute the simulations under different ratios of positive samples and negative samples. Each simulation experiment describes how the validation score changes with respect to the agreements in three classifications. As shown in Figure 4, TAE score has stronger ability of anti-unbalance than Kappa. Figure 4a, 4b and 4c show that the curve changes of Kappa score, accuracy rate, and TAE score with respect to the agreement under different ratios of positive samples and negative samples, respectively. Figure 4d shows that the performances of accuracy rate, Kappa score, and TAE score in the same balanced ratio of positive samples and negative samples, and the ratio is 0.2. Figure 4b shows that accuracy is linear increasing with respect to the agreement. It does not consider the influence of randomness. Figure 4a and Figure 4c show that Kappa score and TAE score are non-linear increasing with respect to the agreement. Both of them consider the influence of randomness. Moreover, from Figure 4a, we can find that Kappa score will be low when the agreement is less than a high threshold about 96% under the setting of an extremely imbalanced label distribution, then the score will increase exploredly when the agreement is above 96%. However, the TAE score will not be influenced by the level of imbalance, just like the accuracy rate. Figure 4d shows that the accuracy rate is higher than Kappa score and TAE score under the situation of the labels distributing balanced. Furthermore, TAE has a similar performance as Kappa. Therefore, TAE can be used to evaluate the validation of our dataset. We point out that the Kappa score means high reliability if it is above 0.6. The corresponding score of TAE is 0.53 with the same agreement and randomness as Kappa whose score is 0.6.

A.2 Methods

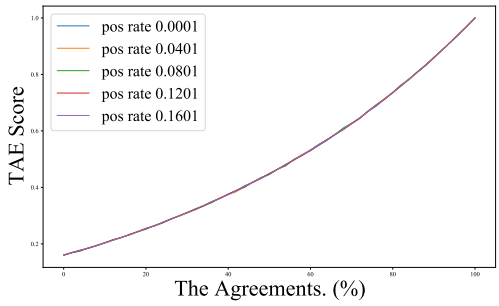
In this section, we provide our method to deal with subtext recognition. The main components are embedding module, feature extracting module, meaning extracting module and comparison module. We



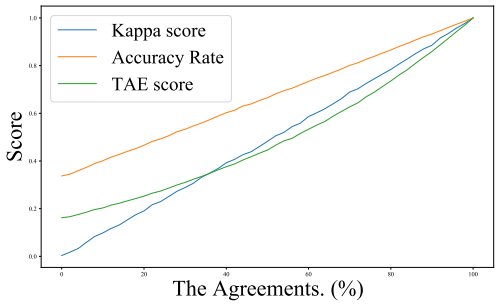
(a) Kappa Changes with Agreements.



(b) Accuracy Changes with Agreements.



(c) TAE Changes with Agreements.



(d) Comparison Of Kappa, Accuracy and TAE.

Figure 4: Validation Metrics Comparison. Pos rate means the ratio of positive samples and all samples.

use GloVe (Pennington et al., 2014), or BERT (Devlin et al., 2019) as embedding module, use LSTM (Hochreiter and Schmidhuber, 1997) as the feature extracting module. Moreover, we use a linear function as meaning extracting module and use softmax as comparison module. The pseudo code is shown in Algorithm 1, where FeaExt is the feature extracting module, Encode is the embedding module, Mean is the meaning extracting module and cls is the classification module. We run the experiment in GTX 1080ti. And we get the results that the precision score is **64.8**, the recall score is **69.9**, the f1 score is **66.1**. We release our code and dataset in `https://anonymous.4open.science/r/ACL-codes-FDC0`. More details can be found in it.

Algorithm 1 Judge whether a sentence is subtext

Input: context sequence c, target sequence s

$e_c \leftarrow \text{Encode}(c)$, $e_s \leftarrow \text{Encode}(s)$

$f_c \leftarrow \text{FeaExt}(e_c)$, $f_s \leftarrow \text{FeaExt}(e_s)$

$m_s \leftarrow \text{Mean}(e_s)$, $m_{cs} \leftarrow \text{Mean}([e_c : e_s])$

$y \leftarrow \text{cls}(f)$, $\quad f \leftarrow [m_s : m_{cs}]$

Output: y $\in \mathcal{R}$