BEYOND WORD-LEVEL TO SENTENCE-LEVEL SENTIMENT ANALYSIS FOR FINANCIAL REPORTS

Chi-Han Du[1] *Ming-Feng Tsai*[2,3] *Chuan-Ju Wang*[1]

[1] Research Center for Information Technology Innovation, Academia Sinica, Taiwan
[2] Department of Computer Science, National Chengchi University, Taiwan
[3] MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan

ABSTRACT

This paper attempts to conduct a sentence-level sentiment analysis with respect to financial risk on a collection of financial reports. Specifically, we first propose a simple yet efficient algorithm to generate financial sentiment phrases (senti-phrases), and then with the obtained senti-phrases, we utilize multiple sentence embedding models for better learning the representations of financial risk sentences. In order to verify the performance of the proposed approach, we conduct a risk classification task of financial sentences on a sentence-level labeled dataset of finance reports. Experimental results show that incorporating the obtained senti-phrases into the embedding-based models improves the classification performance.

***Index Terms*—** Financial NLP, Sentiment Analysis, Sentence-Level Analysis

1. INTRODUCTION

In recent years, the growing demand for analyzing financial texts for better decision strategy has led to the need to push forward the understanding of finance textual information, such as financial reports or news. Due to the prevalence of data analytics, there have been a number of studies on financial text analytics in the fields of finance and computational linguistics [1, 2].

In most of previous studies, the commonly used textual information is the data of companies' annual disclosures. The disclosures are also known as the 10-K filing reports, which are carefully crafted formal documents containing comprehensive information of companies' financial condition. There have been several works on 10-K filing reports [1, 3, 4]. For example, Kogan et al [3] attempt to predict financial risk via the textual information of financial reports using regression approaches, Tsai and Wang [4] propose a learning-to-rank approach for financial risk prediction and evaluating the relations between texts and financial risk. Moreover, Loughran and McDonald [5] propose a finance-specific sentiment lexicon to conduct further financial sentiment analysis. Most of previous studies are based on bag-of-words models or word-level embedding techniques, which usually lead to difficulty in understanding the analyzed results. The difficulty is due to the fact that most finance keywords are context-sensitive according to Liu et al [6].

In order to advance the understanding of financial texts from word level to sentence level, this paper attempts to conduct sentence-level financial sentiment analysis with respect to financial risk on a collection of 10-K financial reports. In particular, we first present a simple yet efficient algorithm to generate financial sentiment phrases (senti-phrases), which are more "meaningful units" and therefore provide more specific implications than words. When counting presence of meaningful units larger than word (or multiword expression, MWE), in average there is only less than 1 MWE per sentence in social web corpus [7], while we had annotated and observed that there are more than 3 MWEs per financial sentence. Before suitable MWE detection models being developed, the proposed algorithm is effcient in terms of time for model training and inference.

We then incorporate the obtained senti-phrases into the sentence embedding models to detect the financial risk sentences, including models based on long short-term memory network (LSTM) [8] or convolutional neural network (CNN) [9], fastText [10], and SiameseCBOW [11]. Experimental results show that the obtained senti-phrases are beneficial to the embedding-based models and further improve the performance.

2. METHODS

2.1. Financial sentiment lexicon

For financial sentiment analysis, Feldman had stated that the lexicon is a crucial resource, usually greatly impacting results and the corresponding analyses [12]. Currently, the six finance-specific word lists proposed by [2] constitute the most reliable and commonly adopted sentiment lexicon in the field of finance. According to the study in [4], the four word lists Fin-Neg, Fin-Pos, Fin-Unc, and Fin-Lit are more relevant to financial risk than the other two lists MW-Strong

(strong modal words) and MW-Weak (weak modal words); hence, this paper only considers the words in these four word lists as seeds while constructing financial senti-phrases. The four lists are shown as follow:

1. Fin-Neg: negative business terminologies (e.g., deficit, default).
2. Fin-Pos: positive business terminologies (e.g., achieve, profit).
3. Fin-Unc: words denoting uncertainty,with emphasis on the general notion of imprecision rather than exclusively focusing on risk (e.g., appear, doubt).
4. Fin-Lit: words reflecting a propensity for legal contest or, per our label, litigiousness (e.g., amend, forbear).

2.2. Financial senti-phrase construction

Definition 1 (Financial senti-phrases) *A financial senti-phrase is a consecutive subsequence of any length $n > 1$ of word tokens $w_1, \ldots, w_n$, such that $\exists i \in \{1 \ldots n\}$ and $w_i \in N \cup P \cup U \cup L$, where the sets of words in Fin-Neg, Fin-Pos, Fin-Unc, and Fin-Lit are defined as N, P, U, and L, respectively.*

We first present a financial senti-phrase construction algorithm to construct meaningful phrases, which is based on the information of rather short n-grams (e.g., bigrams, trigrams) in the corpus. To achieve this goal, we propose the sub-phrase algorithm, which is inspired by the subword algorithm by Sennrich et al [13], to generate reference tables for merging words or word sequences.

Instead of merging frequent pairs of bytes, we merge frequent pairs of word sequences in our algorithm. First, for $n = 2, \ldots, M$, the n-grams of each sentiment lexicon are constructed and counted into a preliminary list, which is defined as frequency table T_M, containing the top k frequent sentiment n-grams and their frequencies. For example, a row in T_M, $t = \{(w_i, w_j, w_k) : 20\}$, denotes a trigram with frequency 20. And the time spent is primarily proportional to k, which means that it is fast to implement.

Note that here we consider only n-grams that include at least one sentiment word from the given sentiment word list. The proposed sub-phrase algorithm then iteratively counts all word pairs and merges each occurrence of the most frequent pair (w_i, w_j) to form a new "word" with an underscore, $(w_i\_w_j)$ (i.e., a sub-phrase). Each merge operation merges two subsequent "word"s into one new "word", updates every pair of "word"s within the frequency table, and records the "word" pair in reference table W, pairs of which are finally used to assemble words in financial texts. For example, the first iteration merges the most frequent word pair *net loss* to *net_loss*; the second iteration merges the most frequent pair *net_loss of* as *net_loss_of*, and so on. More details for the sub-phrase algorithm can be found in Algorithm 1. Once we obtain the reference table W for each sentiment word list, we merge the words in financial texts to construct the corresponding financial senti-phrases. Table 1 gives an example for extracted senti-phrases in the report filed by Rex Energy Corporation in 2013.

2.3. Distributed sentence embedding models

In view of the wide usage of distributed word embedding models, e.g., word2vec, there have been some recent studies focusing on generating effective sentence-level vectors. There are basic two types of approaches for learning sentence embedding vectors: composition-based and context-based approaches. The first type of approaches map individual word vectors to sentence vectors via tangling with a supervised task that depends on class labels, such as models based on LSTM [8] or CNN [9], and fastText [10]. The second type of methods generate generic sentence vectors by using unsupervised approaches, such as SiameseCBOW [11]. In this paper, we adopt the above four models to learn the sentence embedding vectors and as well as incorporate the financial senti-phrases obtained from Algorithm 1 to improve the quality of the learned sentence vectors and thus the classification performance.

3. EXPERIMENTS

3.1. Dataset

In this paper, we use the 10-K corpus[1] to conduct the experiments; the corpus contains 40,708 annual SEC-mandated financial reports on Form 10-K from year 1996 to 2013. Only Section 7 "management's discussion and analysis of financial conditions and results of operations" (MD&A) is utilized in our experiments, in which there are 12,669,628 sentences in total.

The sentence-level risk-labeled dataset is constructed by eight financial specialists including accountants, financial analysts and consultants participated in the annotation task to ensure the quality of the labeling, where 2,432 randomly chosen sentences are labeled as either high (1) or neutral (0) with respect to financial risk. In the annotation process, each of the candidate sentences was labeled by three different annotators, and then the rule of majority was used to determine the degree of risk of the sentence. In total, there are 1,536 and 896 sentences belonging to high-risk and neutral classes, respectively. For both class, we randomly select 243 sentences as the test data, while the remaining sentences in each class are treated as the training data.

[1] `https://cfda.csie.org/10K/data`

Algorithm 1: Sub-phrase algorithm

function Sub-Phrase (T_M, k, ℓ);
Input : A frequency table T_M including the top k most frequent sentiment n-grams and their frequencies, for $n = 2, \ldots, M$; the number of iterations, ℓ
// one row of the table, $t = \{(w_i, w_j, w_k) : 20\}$, denotes a trigram with frequency 20
Output: A reference table, W
$W \leftarrow \{\}$;
for $e \leftarrow 1$ **to** ℓ **do**
  Find the most frequent word pair w_i and w_j in T_M;
  Find all n-grams containing w_i and w_j within T_M;
  Merge these two words into a new “word”;
  Add the merged new “word” $w_i\_w_j$ to the reference table W;
  Delete the most frequent word pair w_i and w_j in T_M;
  Update the frequency table T_M by replacing (w_i, w_j) as $(w_i\_w_j)$;
  // the row $t = \{(w_i, w_j, w_k) : 20\}$ becomes $t^{\text{new}} = \{(w_i\_w_j, w_k) : 20\}$
end
return W;

Rex Energy Corporation / report filing data: March 14, 2013	
Original sentence	Global markets may have a material adverse impact on our business and financial condition that we currently cannot predict.
Sub-phrase approach	Global markets **may_have_a_material_adverse_impact_on** our business and financial condition that we currently cannot predict.

Table 1. An example of generated senti-phrases

3.2. Experimental settings

Experiments are conducted to evaluate the effectiveness of the sentence-level distributed embedding methods and also the extracted financial senti-phrases for the task of sentence-level risk classification. First, without including the information of senti-phrases, we generate sentence-level features from simple to complicated ways by using (1) tf-idf features, (2) the final hidden states of a typical LSTM model, (3) the output vector of the CNN model proposed by Kim [9], (4) fastText, and (5) SiameseCBOW model.

It should be noticed that, for LSTM and CNN, models are being trained only on the labeled sentences, whereas for fastText and SiameseCBOW, the total 12,669,628 sentences are all used to train models and thus generate sentence embeddings. Furthermore, except that fastText trains its own classifier on the labeled sentences, for the rest of four approaches, we adopt the support vector machine classifier (SVM) on the learned sentence-level feature vectors of the labeled sentences to perform the classification task.

Next, we replace part of consecutive words with the senti-phrases generated by our sub-phrase algorithm; after the replacement, new set of feature vectors of sentences is learned by the five approaches.[2] According to the training results of experiments, we set the dimension of sentence vectors as 128 for models based on LSTM and CNN, and those are set to 300 for fastText and SiameseCBOW.

For generating financial senti-phrases, we apply the proposed sub-phrase algorithm on Section 7 (MD&A) of the total 40,708 financial reports. The three parameters, M, k, and ℓ, in the sub-phrase algorithm (see Algorithm 1) are set to 4, 100, and 100, respectively. That is, we calculate the frequencies of sentiment 2-grams, 3-grams, and 4-grams in the MD&A section of financial reports, forming the top k frequency table T_M in Algorithm 1. In the experiments, the 4-fold cross-validation are being adopted for all approaches.

3.3. Experimental results

Table 2 compares the performance of sentence-level financial risk prediction among five different approaches by calculating accuracy and F1 score. It is expected that when the senti-phrases being generated, the corresponding embeddings would become more context-aware, and it is also easier for classification models to distinguish sentences between different risk levels, thus the models being fed with senti-phrases should perform better.

Observe that in general, both fastText and SiameseCBOW achieve better performance than models based on LSTM and CNN, which is due to the fact that the later two models are only trained on the labeled sentences and do not leverage the information provided by other unlabeled sentences. On the other hand, it is observed that the tf-idf baseline also achieves competitive performance; however, it seems that combining words to generate senti-phrases is not beneficial to the traditional bag-of-word model. On the contrary, with

[2]Note that we treat each senti-phrase as a “word” to train the sentence embedding models.

		F1 score	
	Accuracy	High-risk	Neutral
tf-idf	**88.27**	**0.889**	0.876
tf-idf+senti-phrases	87.15	0.883	**0.880**
LSTM [8]	86.96	**0.893**	0.851
LSTM+senti-phrases	**87.14**	0.889	**0.857**
CNN [9]	86.33	0.852	0.891
CNN+senti-phrases	**86.35**	**0.861**	**0.915**
fastText [10]	87.76	0.858	0.895
fastText+senti-phrases	**88.03**	**0.922**	**0.901**
SiameseCBOW [11]	87.92	0.890	**0.902**
SiameseCBOW+senti-phrases	**88.79**	**0.927**	0.888

Table 2. Risk prediction performance

the presence of financial senti-phrases generated by our subphrase algorithm, almost all of the four distributed sentence embedding approaches obtain the improved performance in terms of accuracy and high-risk F1 score. In particular, SiameseCBOW+senti-phrases achieves the best performance in terms of accuracy among the ten models.

4. CONCLUSION

This paper conducts sentence-level financial sentiment analysis with respect to financial risk on a collection of financial statements. We first propose a simple and efficient subphrase algorithm to generate financial senti-phrases, and then with the obtained senti-phrases, we apply different sentence-level distributed embedding models, including models based on LSTM or CNN, fastText, and SiameseCBow, to detect the financial risk sentences. Preliminary experimental results show that the obtained senti-phrases are beneficial to sentence embedding learning models and further improve the performance. The proposed algorithm is fast to compress data and even improve the semantics of NLP models for financial texts, as a result, in the future it could be applied for summarization of financial corpus, or even automatic generation (NLU) for financial reports.

5. REFERENCES

[1] Clemens Nopp and Allan Hanbury, "Detecting risks in the banking system by sentiment analysis," in *Proceedings of EMNLP*, 2015, pp. 591–600.

[2] Tim Loughran and Bill McDonald, "Textual analysis in accounting and finance: A survey," *Journal of Accounting Research*, vol. 54, no. 4, pp. 1187–1230, 2016.

[3] Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith, "Predicting risk from financial reports with regression," in *Proceedings of NAACL*, 2009, pp. 272–280.

[4] Ming-Feng Tsai and Chuan-Ju Wang, "On the risk prediction and analysis of soft information in finance reports," *European Journal of Operational Research*, vol. 257, no. 1, pp. 243–250, 2017.

[5] Tim Loughran and Bill McDonald, "When is a liability not a liability? textual analysis, dictionaries, and 10-ks," *The Journal of Finance*, vol. 66, no. 1, pp. 35–65, 2011.

[6] Yu-Wen Liu, Liang-Chih Liu, Chuan-Ju Wang, and Ming-Feng Tsai, "Fin10k: a web-based information system for financial report analysis and visualization," in *Proceedings of CIKM*, 2016, pp. 2441–2444.

[7] Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta Conrad, and Noah A. Smith, "Comprehensive annotation of multiword expressions in a social web corpus," in *Proceedings of LREC*, 2014, pp. 455–461.

[8] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] Yoon Kim, "Convolutional neural networks for sentence classification," in *Proceedings of EMNLP*, 2014, pp. 1746–1751.

[10] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of EACL*, 2017, pp. 427–431.

[11] Tom Kenter, Alexey Borisov, and Maarten de Rijke, "Siamese cbow: Optimizing word embeddings for sentence representations," in *Proceedings of ACL*, 2016, pp. 941–951.

[12] Ronen Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[13] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," in *Proceedings of ACL*, 2016, pp. 1715–1725.