Numeral Attachment with Auxiliary Tasks

Chung-Chi Chen
Department of Computer Science and Information Engineering,
National Taiwan University, Taiwan
cjchen@nlg.csie.ntu.edu.tw

Hen-Hsen Huang
Department of Computer Science,
National Chengchi University, Taiwan
MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan
hhhuang@nccu.edu.tw

Hsin-Hsi Chen
Department of Computer Science and Information Engineering,
National Taiwan University, Taiwan
MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan
hhchen@ntu.edu.tw

ABSTRACT

In this paper we propose the task of numeral attachment to detect the attached target of a numeral. Compared with other kinds of named entities, numerals provide richer and more crucial information in some domains. Fine-grained understanding of the information embedded in numerals is a fundamental challenge. We develop NumAttach, a pilot dataset for the proposed task based on tweets. Two main challenges of this task include the informal writing style in tweets and the representation of numerals. To address these challenges, we present an embedding technique that considers word and numeral information simultaneously. Furthermore, we design a joint learning model with the capsule network to accomplish the proposed task. We also release NumAttach to the research community as a resource.

CCS CONCEPTS

• **Information systems → Information extraction**.

KEYWORDS

numeral attachment, numeral representation, joint learning

ACM Reference Format:
Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019. Numeral Attachment with Auxiliary Tasks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331361

1 INTRODUCTION

Numerals provide rich and crucial information in documents in many domains. For example, in clinical records, one important piece of information is dosage, expressed by numerals; numerals provide ingredient proportions in recipes; in financial statements, numerals represent up to 17 meanings [2]. Thus fine-grained analysis of numerals is worthwhile. Whereas previous works focus only on predicting exact numbers in documents [10] or on disambiguating the meanings of numerals [2], the relation between a numeral and the target it describes is still obscure.

(T1) *Guess who sold off about $800 million in $MDLZ after losing about $1 billion on $VRX???https://t.co/SHiJutyenv*

One critical operation is the numeral attachment. Taking (T1) as an example, "800 million" is related to $MDLZ, and "1 billion" is associated with $VRX. Note that $MDLZ and $VRX are cashtags, frequently used in financial tweets, standing for Mondelez International, Inc.'s stock and Valeant Pharmaceuticals Interna's stock, respectively. A cashtag is represented as "$" + a stock symbol. In this case, without resolving the relations between cashtags and numerals, it is difficult to compare the two stocks to gauge for instance that the degree of market sentiment for $VRX is higher than $MDLZ [4]. To this end, we define the task of numeral attachment as detecting the attached target (i.e., cashtag) of the numeral and propose a model to perform the task.

Identifying the relation between entities has long been one of the focuses in natural language processing (NLP). However, there still remain unexplored topics, such as the relation between the numeral and the subject it modifies. Although numeral information should be considered a special case when dealing with tokens, little work has been done on this.

The contributions of this paper are threefold. (1) We introduce the task of numeral attachment, in which we detect the attached target of the numeral, and we release NumAttach, an annotated dataset[1]. (2) We investigate representations that are suitable for capturing the meaning in both words and numerals in social media data. (3) We propose a novel joint learning approach to numeral attachment by including auxiliary tasks.

2 RELATED WORK

Many have worked toward fine-grained understanding of financial tweets, which are composed of at least one cashtag [2, 5, 6]. Financial tweets are often the result of investor opinions and analysis; numerals play an important role in these tweets. According to statistics computed from 550K financial tweets, over 83.66% of financial tweets comprise at least one numeral. In this paper, we annotate a dataset and conduct experiments on these financial tweets.

The capsule network architecture has been used for several kinds of tasks involving both images [9] and NLP [11, 12]; it has shown its usefulness especially for classification tasks. To continue this line of research, we propose a capsule network-based joint learning model, and compare its performance with that of state-of-the-art

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331361

[1] http://nlg.csie.ntu.edu.tw/nlpresource/NumAttach/

models. Joint learning models demonstrate state-of-the-art performance in several tasks [3, 7]. It is helpful to learn a main task with auxiliary tasks [8]. In this paper, we develop a novel joint learning architecture toward the proposed task with helpful auxiliary tasks.

To the best of our knowledge, no work has dealt with the problem of numeral attachments in financial social media data, despite the influence and importance of numerals in financial narrative. We compile the NumAttach dataset for extending NLP tasks on financial social media data, and publish the set for academic research.

3 DATASET CONSTRUCTION

There are two kinds of annotations in the NumAttach dataset:

- Main Task: the annotation of the numeral attachment;
- Auxiliary Task: the reason type for the mentioned numeral.

Below, we describe the annotation process. We design the annotation guidelines for annotating the reason for the mentioned numeral. We have publicly released the NumAttach dataset under the CC BY-NC-SA 4.0 license.

3.1 Numeral Attachment

We invited two experts in the financial domain to annotate the attached targets of the numerals. Given a numeral in a tweet, annotators were to choose the attached target of the numeral from all cashtags in the tweet. If no cashtags were related to the target numeral, annotators were to choose "None" as the label. For example, the numeral 65 in (T2) is related to the oil price, and not to the cashtag $NE. As a result, 65 is labeled "None". The other numeral, 8, is the past price of $NE, so the label for 8 is "$NE".

> **(T2)** *$NE OK NE, last time oil was over $65 you were close to $8. Giddy-up...*

To make the dataset more rigorous, only instances for which there was full agreement from all experts are included in NumAttach. Of the 4,847 unique tweets in NumAttach, 70.47% consist of more than one numeral, and 37.76% consist of more than one cashtag. In total, the dataset contains 7,984 instances. Of the numerals, 7,590 out of 7,984 are related to one of the cashtags in the tweet: only 394 numerals are labeled "None". We used 80% of the data for the training set and 20% for the test set. The training and test sets contain 6,403 and 1,581 instances, respectively.

3.2 Numeral Reason Type

We invited an expert from the trading desk of a commercial bank to annotate the reason types for the given numerals in a financial tweets. Note that we sought to use the expert's professional knowledge to improve the performance for detecting the attached target. Firstly, the annotator was to decide whether context could be used to determine why the tweet author mentioned the given numeral. If the reason could be found, the annotator was to choose one of the reason types listed in Table 1. Otherwise, annotator was to choose the label "No Reason" for the numeral. The distribution and examples for each reason are shown in Table 1. For example, (T8) in Table 1 is an example of "Indicator": numeral 143.5, standing for the $SOXL price target, is the result of the investor's analysis, which according to the technical indicator is called DMA. Thus, the annotator chose reason type "Indicator" for 143.5.

4 METHODS

In this paper, we focus on the problem of numeral attachment, and use two auxiliary tasks to improve the performance of the main task: reason detection (reason-binary) and fine-grained reason type classification (reason-type). Both the target numeral and cashtag are given in one instance, and we formulate the problem as a binary classification problem (if the given numeral is related to the given cashtag).

4.1 Text Representation

As shown in Figure 1, the input data, a tweet, is a representation formed by concatenating the token, character, position, and magnitude embeddings. We describe each embedding below.

We collected more than two million financial tweets from Twitter, and learned the document-oriented token embeddings with the skip-gram model. The dimension of the token embedding was 300.

To take into account the informal writing style, we capture out-of-vocabulary (OOV) information with the character embeddings. Both uppercase and lowercase are retained to keep the information that the tweet author seeks to present. Furthermore, as non-alphanumeric symbols are also important in social media data, all punctuation is retained in our character embeddings. This yields a 250-dimensional character embedding for our input representation.

The position embedding is used to show the position of the token in the financial tweet. The longest tweet in the training set consists of 38 tokens.

As the magnitude embedding is designed especially for numerals, it is zero for a word. We label the position of the digits in a numeral to construct the magnitude embedding. For example, the numeral 1.35 in the tweet in Figure 1 is separated into digits (1, 3, 5) and represented as 2, 1, 0, respectively, in the one-hot vector. Tokens that have occurred in the training data are represented by token, character, and position embeddings. OOVs are inevitable, especially when dealing with social media data; we represent such words using character embeddings. To retain numeral information during encoding, in this work, we use character, position, and magnitude embeddings. In Section 5 we analyze the usefulness of each embedding in our representation.

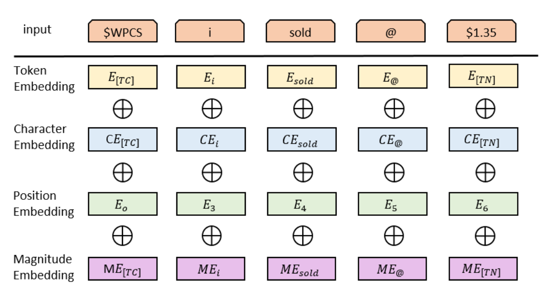

Figure 1: Input representation. The input embedding is the concatenation ($\bigoplus$) of the token, character, position, and magnitude embeddings.

Table 1: Distribution (left) and examples (right) for auxiliary task

Reason type	Occurrence	Reason	Numeral		Example tweet
Asset	106	stake	700	**(T3)**	$NL has $**700**m in $KRO stake.
Liability	22	bonds	49	**(T4)**	$WIN bought back $**49**mm worth of bonds and **9.1**mm shares in the quarter.
Equity	32	shares	9.1		
Income	116	revenue	500	**(T6)**	$WATT $BRCM is contact, revenue $**500**M (Apple alone).
Economics	10	GDP	2.3	**(T7)**	BofA cuts Q4 GDP from **2.3**% to **1.8**% on surge in trade deficit $GLD $SLV
Indicator	280	dma	143.5	**(T8)**	$SOXL 133.5 entry, stop loss 130, target 50 dma **143.5**.
Pattern	111	Reversal candle	342.5	**(T9)**	$IBB had to ride this Train with Reversal candle. Down more than 9% from **342.50**.
No Reason	7,307		2	**(T10)**	$AMD OMG where was all these people when $amd price was **2** us dollars ???

4.2 Joint Model

The input of our model is a tweet with n tokens $\boldsymbol{x} = \{x_1, x_2, x_3, ..., x_n\}$ and a numeral-cashtag pair composed of the target numeral and the target cashtag. We provide their positions i and j, where $1 \leq i, j \leq n$ and $i \neq j$. The target x_i can appear either before or after the target cashtag x_j. The numeral and cashtag separate the input tweet into three parts:

- **PrecedingContext**: Context preceding both the numeral and the cashtag is denoted as $\{x_1, x_2, ..., x_{\min(i,j)-1}\}$.
- **MiddleContext**: Context between the numeral and the cashtag is denoted as $\{x_{\min(i,j)+1}, ..., x_{\max(i,j)-1}\}$.
- **FollowingContext**: Context following the numeral and the cashtag is denoted as $\{x_{\max(i,j)+1}, ..., x_{n-1}, x_n\}$.

The language model for each type of context is learned independently. We extract the features via a convolutional neural network (CNN). In the capsule network, we adopt a squashing function [9] to shrink the feature matrix. We learn the capsule representation by sharing weights between child and parent layers [13]. There are N_t vectors in the output of the capsule network, where N_t stands for the number of classes of each task; this is 2, 2, and 8 for our main task and the two auxiliary tasks, respectively. After this the bidirectional-GRU (Bi-GRU) is used to capture the features in both directions. The features from the latest node of Bi-GRU serve as the representation of the context. Thus we have the information in PrecedingContext, MiddleContext, and FollowingContext represented as h_{pre}, h_{mid}, and h_{fol}, respectively. Finally, the concatenation of the three representations is used by our main task and the two auxiliary tasks. The output layers h_{Main}, h_{auxBin}, and h_{auxTyp} are activated with the softmax function.

The loss of each task, ℓ_t, is calculated by the following objective function:

$$\ell_t = \frac{1}{N_t} \sum_{k=1}^{N_t} y_k^t \max(0, \alpha - \hat{y}_k^t)^2 + \lambda(1 - y_k^t) \max(0, \hat{y}_k^t - (1-\alpha))^2, \tag{1}$$

where N_t is the number of instances in task t, y_k^t is the actual label of the instance k for the task, and $\hat{y}_k^t$ is the predicted label of the instance k for the task. We follow the recommendation of previous work [9] to set α to 0.9 and λ to 0.5. Finally, the overall loss is the weighted sum of all ℓ_t: $L = \sum_t^T w_t \ell_t$, where w_t is the weight of each task. We set the weight for the main task to 0.8. For each of the auxiliary tasks, a weight of 0.1 is set. In our model, weighted loss fine-tunes the parameters for the main task with information from the auxiliary tasks. In Section 5 we discuss the improvement provided by the auxiliary tasks.

The remaining hyperparameters of our model are listed as follows. The CNN filter size is 256, the output dimension of each capsule network is 16, the hidden dimension size of Bi-GRU is 32, and the dimensions of h_{pre}, h_{mid}, and h_{fol} are 300.

5 EXPERIMENTS

5.1 Overall Performance

We adopt a state-of-the-art relation extraction model as our baseline and develop an attentive-CNN model with our joint learning architecture for comparison. In addition, the result of logistic regression with bag-of-words features is also reported for comparison.

- Adversarial training (AT): the state-of-the-art model for relation extraction [1], which adds some perturbations while training the model constructed by bidirectional LSTM and CRF.
- Attentive-CNN: We add an attention layer as the first layer of the capsule network model, and replace the capsule network in our model with a two-layer CNN.

Table 2: Experimental results of numeral attachment

Model	Macro-F1
Logistic regression	51.11%
AT [1]	53.36%
Attentive-CNN	72.64%
Capsule-based	**73.46%**

The experimental results are shown in Table 2. We use the macro-F1 score to evaluate the performance of the models. The capsule-based model yields the best performance, and the attentive-CNN model underperforms the capsule-based model by only 0.82%, which is insignificant under McNemar's test ($p > 0.05$). Both models outperform the state-of-the-art model in the numeral attachment task, while attests the success of the proposed joint learning architecture.

5.2 Discussion

Table 3 shows the evidence for the usefulness of the auxiliary tasks. If we train the capsule-based model without any auxiliary task, the performance is 67.14%, which is 2.83% worse than the model trained with the coarse-grained auxiliary task. Co-training with both auxiliary tasks improves the performance by 3.49% over the

Table 3: Ablation analysis of auxiliary tasks

Model	Caps-m	Caps-mb	Caps-mt	Caps-all
Main task	v	v	v	v
Reason-binary		v		v
Reason-type			v	v
Macro-F1	67.14%	69.97%	66.95%	73.46%

Table 4: Ablation analysis of input representation

Model	Caps-w	Caps-wc	Caps-wcp	Caps-all
Token	v	v	v	v
Character		v	v	v
Position			v	v
Magnitude				v
Macro-F1	60.08%	69.59%	69.73%	73.46%

model co-trained with the coarse-grained task only. Overall, the model co-trained with two auxiliary tasks is significantly superior to other settings under McNemar's test with $p < 0.05$. These results confirm the effectiveness of the auxiliary tasks proposed in our approach.

Although the Caps-mt model, which is co-trained only with the reason-type auxiliary task, does not yield performance significantly different from that of the Caps-m model under McNemar's test, when comparing the Caps-mb and Caps-all models, adding the reason-type auxiliary task improves the performance of the reason-binary auxiliary task, and further advances the performance of the main task.

To show the influence of different embeddings, we investigated the modified input representations with the capsule-based model. In Table 4, Caps-w denotes the model with only token embeddings as features, Caps-wc that with token and character embeddings, Caps-wcp that with token, character, and position embeddings, and Caps-all that with token, character, position, and magnitude embeddings. The results show the model with token and character embeddings outperforms the model with only token embeddings by 9.51%. Position embeddings have little influence on the proposed task. With the magnitude embeddings, the performance further improves by 3.73% and is significantly superior to the model without magnitude embeddings under McNemar's test.

The performance of the reason-binary and reason-type auxiliary tasks is 59.40% and 15.41%, which is 11.75% and 3.5% higher than the majority guess. These results illustrate that the models are actually learning something useful during model training.

6 CONCLUSION

In this paper, we present numeral attachment, an important task when dealing with numerals in social media data. We propose a novel joint learning approach to complete this task. Related auxiliary tasks are carefully chosen based on our observation of a dataset annotated with expertise knowledge. A number of representations for the input data are also explored. Experimental results confirm the effectiveness of our approach, and an ablation analysis is conducted and discussed.

In future work, we plan to apply our approach to capture numeral attachment in application scenarios and domains such as clinical, sport, and geographic documents, where numerals play important roles.

ACKNOWLEDGMENTS

This research was partially supported by the Ministry of Science and Technology, Taiwan, under grants MOST 107-2218-E-009-050-, MOST-106-2923-E-002-012-MY3, MOST-107-2634-F-002-011-, and MOST-108-2634-F-002-008-, and by Academia Sinica, Taiwan, under grant AS-TP-107-M05.

REFERENCES

[1] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2830–2836. https://www.aclweb.org/anthology/D18-1307

[2] Chung-Chi Chen, Hen-Hsen Huang, Yow-Ting Shiue, and Hsin-Hsi Chen. 2018. Numeral Understanding in Financial Tweets for Fine-grained Crowd-based Forecasting. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 136–143.

[3] Ying Chen, Wenjun Hou, Xiyao Cheng, and Shoushan Li. 2018. Joint Learning for Emotion Classification and Emotion Cause Detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 646–651. https://www.aclweb.org/anthology/D18-1066

[4] Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 519–535. https://doi.org/10.18653/v1/S17-2089

[5] Shijia E., Li Yang, Mohan Zhang, and Yang Xiang. 2018. Aspect-based Financial Sentiment Analysis with Deep Neural Networks. In *Companion Proceedings of the The Web Conference 2018 (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1951–1954. https://doi.org/10.1145/3184558.3191825

[6] Hitkul Jangid, Shivangi Singhal, Rajiv Ratn Shah, and Roger Zimmermann. 2018. Aspect-Based Financial Sentiment Analysis Using Deep Learning. In *Companion Proceedings of the The Web Conference 2018 (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1961–1966. https://doi.org/10.1145/3184558.3191827

[7] Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint Learning for Targeted Sentiment Analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 4737–4742. https://www.aclweb.org/anthology/D18-1504

[8] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[9] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*. 3856–3866.

[10] Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 2104–2115. https://www.aclweb.org/anthology/P18-1196

[11] Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. 2018. Sentiment Analysis by Capsules. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1165–1174. https://doi.org/10.1145/3178876.3186015

[12] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot User Intent Detection via Capsule Neural Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3090–3099. https://www.aclweb.org/anthology/D18-1348

[13] Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating Capsule Networks with Dynamic Routing for Text Classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3110–3119. https://www.aclweb.org/anthology/D18-1350