Document Classification with Word Sense Knowledge

Anonymous ACL submission

Abstract

The performance of Word Sense Disambiguation (WSD) on a standard evaluation framework has reached an estimated upper bound. However, there is limited research on the application of WSD to relevant NLP tasks due to the high computational cost of supervised systems. In this paper, we propose a partial WSD method with sense category information and incorporate the sense knowledge into a supervised document classification framework. Experimental results show that the proposed method can constantly boost the system's performance on document classification datasets against strong baselines.

1 Introduction

Text classification is one of the primary tasks in NLP community. A wide range of methods have been proposed to tackle the task, including traditional methods (Androutsopoulos et al., 2000; Tan, 2006; Forman, 2008), currently prevailing deep learning architectures (Kim, 2014; Zhang et al., 2016; Peters et al., 2018) and also graph neural networks (Yao et al., 2019; Huang et al., 2019; Zhang et al., 2020). The newly proposed methods can obtain outstanding performance on standard text classification datasets.

In the supervised category, most of the previous work focuses on learning the relatively shallow mapping between the vector representation of the provided text and its label, rarely considering the senses behind the words. In many tasks of text classification, systems are required to distinguish which domain the given text is covering. In many cases, the difficulty of selecting a text's major domain originates from the ambiguity of the words. For example, the word 'court' might appear in the domain of 'sport' or 'law'. The sense knowledge of each sense, *court.n.04 {a specially marked horizontal area within which a game is played}* and *court.n.07 {a tribunal that is presided over by a magistrate or by one or more judges who administer justice according to the laws}*, can assist the assignment of the text's domain.

Among the limited research on the contribution of Word Sense Disambiguation (WSD) to text classification, most requires explicit disambiguation of words (Hung and Chen, 2016; Sinoara et al., 2018; Shimura1 et al., 2019). Although the proposed approaches can elevate the systems' performance on text classification datasets, the explicit disambiguation of words in the given text leads to low efficiency, especially for the classification of documents. Further, the fine-grained disambiguation of words is somehow redundant since many text categorization tasks only require coarse-grained genre information.

In this paper, we propose a method to partially disambiguate the words in a given document to retrieve the necessary category information of the text. The disambiguation is implemented with a coarse sense inventory (CSI, Lacerra et al., 2020), using a majority voting mechanism. The retrieved sense knowledge (sense definition) of disambiguated words is then incorporated into a supervised text classification architecture. Experimental results on five document categorization datasets have shown the effectiveness of the proposed method. We summarize the contribution of our approach as follows:

(1) We propose a fast and efficient partial WSD method to retrieve necessary category information for downstream text categorization tasks. On a proportion of standard WSD datasets, the method extensively outperforms a strong baseline.

(2) We propose a sense knowledge incorporation method in a supervised text

classification architecture, which constantly raises the system's performance on five document classification datasets.

2 Method

In this section, we will first briefly introduce the task of WSD before illustrating the proposed partial WSD of a document. Then, we will explain the incorporating method of the obtained sense knowledge in a supervised text classification framework.

2.1 WSD

WSD is to select the correct sense of a word in its context. Candidate senses are from a sense inventory such as WordNet. For example, in the sentence "Players had to reserve a court in advance", the correct sense of 'court' is *court.n.04 {a specially marked horizontal area within which a game is played}*. In WordNet 3.1, 'court' has 11 meanings and many of them are excessively fine-grained for relevant NLP tasks including text categorization.

In CSI, most of WordNet senses are categorized into 45 coarse-grained classes including 'TRANSPORT & TRAVEL', 'PHYSICS & ASTRONOMY' and 'MUSIC, SOUND & DANCING'. For instance, *car.n.01* is mapped to 'TRANSPORT & TRAVEL'. These coarse-grained labels can adequately convey domain knowledge for text classification. We utilize these labels to conduct partial WSD of documents.

2.2 Partial WSD

In the scenario of document classification, the fine-grained WSD is somehow unnecessary, especially utilizing currently complex supervised WSD architectures (Bevilacqua and Navigli, 2020, Blevins and Zettlemoyer, 2020). On the contrary, we employ a majority voting mechanism to obtain the necessary sense knowledge for deciding the domain of a document.

For each noun word w_i in a document D, each of its potential senses $s_{i,k} \epsilon S_{w_i}$ is retrieved from WordNet. For each sense $s_{i,k}$, we score its corresponding coarse-grained labels l in CSI by frequency and consider the most frequent coarse-grained label as the major label L. Formula (1) demonstrates the detailed calculation of each CSI label's score. I_Δ is an indicative function where it returns 1 if a particular label l in CSI is linked to sense $s_{i,k}$ and returns 0 otherwise. Here, we only use nouns to capture the document domain.

$$L = \underset{l \in CSI}{\operatorname{argmax}} \sum_{w_i \in D} \sum_{s_{i,k} \in S_{w_i}} I_\Delta(l \in CSI(s_{i,k})) \quad (1)$$

Here, we conjecture that the CSI label L is the major domain that the document belongs to. Then, we retrieve all the senses in D that are linked to the major domain label L in CSI and incorporate the knowledge (definition) of these senses into the supervised text classification framework.

2.3 Text Classification Framework

In our baseline model, the classification is implemented by fine-tuning a pre-trained language model (PLM). Precisely, we first input the text sequence into a PLM and retrieve the encoded features. Then, the features are mapped into a vector whose dimension is the number of classes for the text classification. After a SoftMax function is applied to the vector, a cross-entropy loss is computed against the processed vector and the ground-truth distribution. We take BERT (Devlin et al., 2019) as an example, for each input text sequence x_i, its representation is from BERT's last layer at [CLS] position, shown in formula (2), on which a feed-forward network and a SoftMax function are applied. The cross-entropy loss is calculated with formula (3), where y_i is the ground-truth distribution.

$$v_{x_i} = BERT_{-1}^{CLS}(x_i) \quad (2)$$

$$\mathcal{L}(x_i, y_i) = -\sum_{k=1}^{|y_i|} y_{i,k} log(softmax(mlp(v_{x_i}))^k) \quad (3)$$

Here, we augment the text representation with sense knowledge. Specifically, for each text sequence x_i, the above partial WSD returns a series of senses S_L that are linked to the domain label L. To obtain the sense representation V_{S_L}, we also utilize a PLM to encode all senses' WordNet definition in a batch, $def(S_L)$, using the last layer's output at [CLS] position, demonstrated in formula (4). To avoid high computational expense, this PLM is frozen during training.

$$V_{S_L} = BERT_{-1}^{CLS}(def(S_L)) \quad (4)$$

We utilize a self-attention layer to obtain a context-aware sense representation as in formula (5) and (6), similar to the implementation in Yu and Jiang (2019). W_Q, W_K and W_V are learnable parameters. d and m are respectively the dimension of PLM hidden states and the number of heads in the self-attention layer. Unlike the setting in transformer encoder, we utilize V_x as the query to calculate the weights for each sense representation in V_{S_L}. As in formula (7), V_x is the sum of BERT's output at the last layer in all positions but [CLS], with x_i being its input.

$$v_{S_L} = [v_{1,S_L}, v_{2,S_L}, \dots, v_{m,S_L}] \quad (5)$$

$$v_{k,S_L} = softmax(\frac{[W_{Q_k} V_x]^T [W_{K_k} V_{S_L}]}{\sqrt{d/m}})[W_{V_k} V_{S_L}]^T \quad (6)$$

$$V_x = \sum_{j=1}^{|x_i|} BERT_{-1}^{j}(x_i) \quad (7)$$

Similar to BERT, we apply a feed-forward network (mlp) and two-layer norms (LN) with residual connections to the multi-head self-attention layer's output to obtain the context-aware sense representation, shown in formula (8).

$$v_{S_L} = LN(LN(V_x + v_{S_L}) + mlp(LN(V_x + v_{S_L}))) \quad (8)$$

$$v_{xs} = [v_{x_i}, v_{S_L}] \quad (9)$$

The context-aware sense representation v_{S_L} is then concatenated with the original context representation v_{x_i} to obtain the two-way representation v_{xs}, as in (9). Then, similar to the implementations in formula (5), (6) and (8), we utilize another encoder to fuse the two-way representation. One different implementation is that we use v_{xs} as the query in the multi-head self-attention layer. We then utilize a feed-forward network and a SoftMax function to transform the fused vector and compute the loss as in formula (3).

3 Datasets and Settings

3.1 WSD datasets

In order to evaluate the performance of the partial WSD method, we utilize a standard WSD evaluation framework (Raganato et al., 2017) which contains five all-words WSD datasets. For comparison, we also implement a hard-to-beat baseline for knowledge-based methods. For any given word, the baseline selects the WordNet 1st sense as its prediction. We note that the partial WSD method only disambiguates a proportion of the words in the given document. We therefore compare the method's performance only on those disambiguated instances. We note that WordNet 1st sense is a high-quality knowledge derived from a sense-annotated corpus.

Dataset	ALL	Train	Test	Class	A.Len
20NG	18,846	11,314	7,532	20	221.26
R8	7,674	5,485	2,189	8	65.72
R52	9,100	6,532	2,568	52	69.82
AGnews	127.6k	120k	7.6k	4	135.82
DBPedia	630k	560k	70k	14	46.13

Table 1: Document Classification Datasets

3.2 Text Classification Datasets

For text classification datasets, we select five English document classification tasks including 20NG[1], reuters-8 (Lewis et al., 2004), reuters-52, AGnews (Zhang et al., 2015) and DBpedia (Zhang et al., 2015). The statistics for these datasets are shown in Table 1. The average length (A.Len) of the documents in the datasets is relatively large, especially for 20NG and AGnews.

3.3 Experiment Setting

To perform a fair comparison between our Sense-Aware Text Classification framework and the baseline, we implement a 10-fold cross-validation experiment on the datasets. For small datasets (20NG, reuters-8, reuters-52), we combine the train and test set and apply a random split with a ratio of 0.75:0.25. For the other two datasets (AGnews and DBpedia), we randomly sample 30,000 instances from the combined dataset and apply the same split.

The baseline is detailed in formula (2) and (3). For comparison, we also implement a system that incorporates WordNet 1st sense knowledge of the words in the given text. It is noteworthy that the number of retrieved senses in each document might be excessively large. To lower the computational cost, we only use the first 32 senses in S_L according to the word order, for the sense knowledge incorporation. The detailed hyper-parameters for the model are shown in table 2.

4 Results

	20NG	R8/R52/…
lr	1.00E-06	1.00E-05
warmup	0.1*total_step	0.1*total_step
batch-size	4	16
epoch	40	10
sense-num	32	32
max-seq-len	512	256

Table 2: Model Hyper-parameters

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

Label	CSI Label	Text
Business	BUSINESS_ ECONOMICS_ AND_ FINANCE_	Tearaway world oil prices, toppling records and straining wallets, present a new economic menace barely three months before the US presidential elections.
Sports	SPORT_ GAMES_ AND_ RECREATION_	The Cleveland Indians pulled within one game of the AL Central lead, scoring four runs in the first inning and beating the Minnesota Twins 7-1 Saturday night behind home runs by Travis Hafner and Victor Martinez.
World	LAW_ AND_ CRIME_	Thousands of Palestinian prisoners in Israeli jails began a hunger strike for better conditions Sunday, but Israel 's security minister said he didn't care if they starved to death.
Science and Technology	BIOLOGY_	Three shark attacks off the Texas coast in the past two months are unusual but don't mean there are more sharks than normal along the beach or that they are getting bolder, marine biologists and other experts say.

Table 4: PWSD Examples from AGnews

	All Instances	PWSD Instances	WN 1st	PWSD
SE2	2282	612	0.686	**0.693**
SE3	1850	554	0.646	**0.673**
SE07	455	91	**0.516**	0.407
SE13	1644	740	0.578	**0.791**
SE15	1022	417	0.619	**0.753**
ALL	7253	2414	0.626	**0.718**

Table 3: PWSD Performance

4.1 Partial WSD

Table 3 demonstrates how the proposed partial WSD method and the WordNet 1st baseline perform on a standard WSD evaluation framework including five separate datasets (SE2, SE3, SE07, SE13 and SE15) and their combination (ALL). The 'ALL Instances' column indicates the number of sense-annotated instances in each dataset. The latter column shows the number of instances that the proposed method manages to make a prediction. The last two columns report the systems' performance on 'PWSD Instances'.

It is revealed that although PWSD can merely disambiguate one third of the instances, it obtains an overwhelming advantage on these instances, surpassing the baseline by 14.7%. The margins are even larger on SE13 (36.7%) and SE15 (21.7%), which contains documents from 13 domains and 4 domains respectively. In contrast, PWSD performs poorly on SE07 even though it can only disambiguate 20% of the labelled words. On average, the ambiguity of this dataset is extensively larger than the others, since it only labels those more ambiguous words while discarding the others.

4.2 Text Classification

Table 4 demonstrates some PWSD examples from the AGnews dataset, providing evidences that PWSD can capture the major domain information of a given text. Further, for coarse-grained document labels, PWSD can even detect fine-grained domains, shown in the last two rows in the table. For instance, the label for the third example is a coarse-grained label, 'world'. PWSD manages to detect its fine-grained label from CSI, 'LAW_AND_CRIME_', which is precisely what the text covers.

	20NG	R8	R52	AG	DBP
Baseline	0.834	0.956	0.890	0.871	0.956
WN 1st	0.858	**0.965**	**0.931**	0.870	**0.959**
PWSD	**0.863**	0.964	0.929	**0.874**	**0.959**

Table 5: Document Classification Performance

Table 5 shows different systems' performance on 5 document classification tasks. It reveals that the proposed sense-aware framework constantly outperforms the baseline. Also, the gap becomes larger if the task (20NG and R52) becomes more difficult (longer documents and more classes). It is worth mentioning that directly incorporating the WordNet 1st sense knowledge slightly outperforms the system that employs the senses from PWSD in many cases. However, PWSD only relies on less expensive resources than WordNet 1st sense, which is more portable to multilingual scenarios.

5 Conclusion

In this paper, we propose a simple partial WSD method and incorporate the disambiguated senses' knowledge into a supervised text classification framework. Experiments have shown the effectiveness of the partial WSD, obtaining extensively higher performance on domain-specific datasets. Moreover, the proposed sense-aware text classification framework constantly outperforms the baseline on five document classification datasets.

6 Ethics Impact Statement

This paper does not involve the presentation of a new dataset, an NLP application and the utilization of demographic or identity characteristics in formation.

References

Ion Androutsopoulos, John Koutsias, Konstantinos V Chandrinos, George Paliouras, and Constantine D Spyropoulos. 2000. An evaluation of naive Bayesian anti-spam filtering. arXiv preprint cs/0006013

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. *In ACL 2020*, pages 1006-1017. Association for Computational Linguistics.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: raising the state of the art in word sense disambiguation by incorporating knowledge graph information. *In ACL 2020*, pages 2854-2864. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *In NAACL 2019*, pages 4171-4186, Minneapolis, Minnesota.

George Forman. 2008. Bns feature scaling: an improved representation over tf-idf for svm text classification. In CIKM. ACM

Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. In EMNLP.

Chihli Hung, Shiuan-Jeng Chen. 2016. Word sense disambiguation based sentiment lexicons for sentiment classification. *Knowledge-Based Systems*. Volume 110, Pages 224–232.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 1746–1751.

Caterina Lacerra, Michele Bevilacqua, Tommaso Pasini, Roberto Navigli. 2020. CSI: a coarse sense inventory for 85%word sense disambiguation. *In Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*, Pages 8123-8130. Association for the Advancement of Artificial Intelligence.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research (JMLR), 2004.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In NAACL-HLT, pages 2227–2237.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. *In EACL 2017*, pages 99-110, Valencia, Spain.

Roberta A. Sinoara, Jose Camacho-Collados, Rafael G. Rossi, Roberto Navigli and Solange O.Rezende. 2018. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*. Volume 163, Pages 955-971.

Kazuya Shimura, Jiyi Li and Fumiyo Fukumoto. 2019. Text Categorization by Learning Predominant Sense of Words as Auxiliary Task. In *ACL 2019*. Pages 1109–1119. Association for Computational Linguistics.

Songbo Tan. 2006. An effective refinement strategy for KNN text classifier. Expert Systems with Applications

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In AAAI 2019. Association for the Advancement of Artificial Intelligence.

Jianfei Yu and Jing Jiang. Adapting BERT for Target-Oriented Multimodal Sentiment Classification. in *IJCAI 2019*, Pages 5408-5414. Association for the Advancement of Artificial Intelligence.

Rui Zhang, Honglak Lee, Dragomir R. Radev. 2016. Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents. In Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies, pages 1512–1521.

Xiang Zhang, Junbo Zhao, Yann LeCun. 2015. Character-Level Convolutional Networks for Text Classification. In Advances in Neural Information Processing systems, pages 649–657.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, Liang Wang. 2020. Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks. In ACL, pages 334–339.