KHAN: Knowledge-Aware Hierarchical Attention Networks for Accurate Political Stance Prediction

Yunyong Ko
yyko@illinois.edu
University of Illinois at
Urbana-Champaign, IL, USA

Seongeun Ryu
ryuseong@hanyang.ac.kr
Hanyang University
Seoul, Republic of Korea

Soeun Han
sosilver@hanyang.ac.kr
Hanyang University
Seoul, Republic of Korea

Yeongseung Jeon
ysj@g.ucla.edu
University of California
Los Angeles, CA, USA

Jaehoon Kim
jaehoonkimm@hanyang.ac.kr
Hanyang University
Seoul, Republic of Korea

Sohyun Park
sally5004@ajou.ac.kr
Ajou University
Suwon, Republic of Korea

Kyungsik Han
kyungsikhan@hanyang.ac.kr
Hanyang University
Seoul, Republic of Korea

Hanghang Tong
htong@illinois.edu
University of Illinois at
Urbana-Champaign, IL, USA

Sang-Wook Kim[*]
wook@hanyang.ac.kr
Hanyang University
Seoul, Republic of Korea

ABSTRACT

The political stance prediction for news articles has been widely studied to mitigate the *echo chamber* effect – people fall into their thoughts and reinforce their pre-existing beliefs. The previous works for the political stance problem focus on (1) identifying political factors that could reflect the political stance of a news article and (2) capturing those factors effectively. Despite their empirical successes, they are not sufficiently justified in terms of how effective their identified factors are in the political stance prediction. Motivated by this, in this work, we conduct a user study to investigate important factors in political stance prediction, and observe that the *context* and *tone* of a news article (*implicit*) and *external knowledge* for real-world entities appearing in the article (*explicit*) are important in determining its political stance. Based on this observation, we propose a novel knowledge-aware approach to political stance prediction (**KHAN**), employing (1) hierarchical attention networks (HAN) to learn the relationships among words and sentences in three different levels and (2) knowledge encoding (KE) to incorporate external knowledge for real-world entities into the process of political stance prediction. Also, to take into account the subtle and important difference between opposite political stances, we build two independent political knowledge graphs (KG) (i.e., KG-lib and KG-con) by ourselves and learn to fuse the different political knowledge. Through extensive evaluations on three real-world datasets, we demonstrate the superiority of KHAN in termss of (1) accuracy, (2) efficiency, and (3) effectiveness.

CCS CONCEPTS

• **Information systems → Web mining**; • **Computing methodologies → Neural networks**; • **Human-centered computing**;

KEYWORDS

political stance prediction, echo chamber effect, hierarchical attention networks, knowledge graph embedding

ACM Reference Format:
Yunyong Ko, Seongeun Ryu, Soeun Han, Yeongseung Jeon, Jaehoon Kim, Sohyun Park, Kyungsik Han, Hanghang Tong, and Sang-Wook Kim. 2023. KHAN: Knowledge-Aware Hierarchical Attention Networks for Accurate Political Stance Prediction. In *Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3543507.3583300

1 INTRODUCTION

With the prevalence of web-based news platforms, people are having more chances to obtain a variety of high-quality information about social and political issues. In general, people tend to prefer news contents which have similar political stances to them [38]. For example, people who have a conservative stance often prefer news articles from conservative news media such as Fox and Breibart, while those with a liberal stance might prefer articles from liberal news media such as CNN, New York Times, and Washington Post. As this tendency becomes intensified, however, people could be trapped in their own opinions and reinforce their pre-existing beliefs by limiting exposure to other news articles having different opinions. This is called the *echo chamber* effect [60], a fundamental reason that leads to serious social polarization by hindering positive and active communications among people [7, 9, 14, 15, 22, 68]. In addition, machine learning (ML)-based recommendation systems have been exacerbating this problem since their goal is to recommend news articles that are likely to be preferred by users [2, 16, 28, 57].

Many studies have been proposed to mitigate the echo chamber effect [1, 19, 37, 45, 46, 52, 53]. They mainly focus on exposing *diverse* opinions (e.g., news articles with different stances) to people in order to prevent them from falling into their existing beliefs too

[*] Corresponding author

WWW '23, May 1–5, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583300

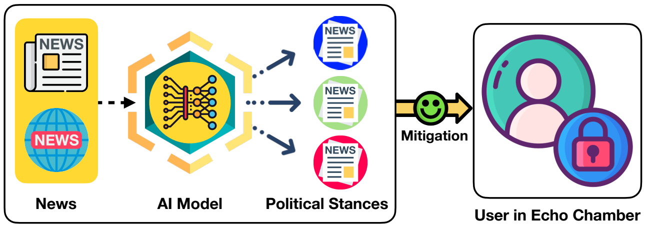


Figure 1: Accurate provision of diverse political stances to mitigate the echo chamber effect.

much. In computer science, for example, many groups of researchers have studied the *diversification* of news recommendation systems in order to provide news articles not only interesting but also fresh to users [23, 49, 72, 74]. Through this effort, people could view a given issue from various perspectives and understand how rational or biased they are, thereby leading to the mitigation of the echo chamber effect [10]. Based on this understanding, from the technical aspect, it is critical to accurately predict the political stance of a given news article (i.e., *political stance prediction*), which this paper focuses on. This is because the accurate provision of news articles with different political stances would not only (1) have people experience diverse political stances but also (2) allow researchers to investigate ways to support news consumption in a balanced standpoint from people's experiences and behaviors [45]. Although there has been much effort to predict political stances of news articles by domain experts, it encompasses a manual process of determining political stances that requires a great amount of time and effort and may be also influenced by human biases [26].

For this political stance prediction, existing language models such as GloVe [55], BERT [29], and RoBERTa [48] can be applied. However, it is reported that a single application of language models empirically failed to achieve high accuracy in the political stance prediction because they are not designed specially for predicting the political stance of a news article [13, 44, 69]. Recently, in order to overcome the limitation of general language models, deep neural networks (DNN) models for political stance prediction have been proposed [13, 42, 44, 69]. They identify social, linguistic, and political factors reflecting the political stance of a news article and design new model architectures to effectively capture the identified factors. Although these DNN models empirically achieved higher accuracies than existing language models, they are not sufficiently justified in terms of why and how effective their identified factors are in predicting the political stances of news articles.

From this motivation, we first conducted a user study in order to investigate what factors real-world users consider and how important the factors are in determining the political stances of news articles. We provided six articles with different political stances and carefully-chosen factors to 136 respondents via Amazon Mechanical Turk. We asked them to respond how important each factor is in their decision with a scale [1:(not at all)-5:(very much)][1]. Through the user study, we observe that the "context" of a new article is the most important factor in deciding its political stance, followed by keyword, person, tone, and frequently used word. This result gives us an important lesson: it is crucial (1) *to learn the relationships among words/sentences to capture the context and tone of a news article*, which is *implicitly* reflected in the article, and (2) *to understand the interpretation and sentiment to real-world entities (e.g., keyword and person)*, which *explicitly* appear in a news article.

Towards reflecting both the explicit and implicit factors, in this paper, we propose a novel approach to accurate political stance prediction, named **K**nowledge-aware **H**ierarchical **A**ttention **N**etworks (**KHAN**). KHAN consists of two key components: (1) hierarchical attention networks (HAN) to learn the relationships among words/sentences in a news article with 3-level hierarchy (i.e., word-level, sentence-level, and title-level), rather than learning the entire article itself, and (2) knowledge encoding (KE) to incorporate both common and political knowledge for real-world entities, necessary for understanding a news article, into the process of predicting the political stance of a news article. Regarding political knowledge, the interpretation and sentiment to even the same entities can be different depending on its political stance [19, 45, 52]. To address the subtle but important difference, we construct two knowledge graphs (KG) with different political stances, KG-lib and KG-con, and design KE to learn to fuse the information extracted from the political knowledge graphs. To our best knowledge, this is the first work that leverages both common and political knowledges separately, further reflecting the different political knowledges.

The main contributions of this work are as follows.

- **Datasets**: To reflect the different political knowledge of each entity, we build two political knowledge graphs, KG-lib and KG-con. Also, for extensive evaluation, we construct a large-scale political news datatset, AllSides-L, much larger (48×) than the existing largest political news article dataset.[2]
- **Algorithm**: We propose a novel approach to accurate political stance prediction (KHAN), employing (1) hierarchical attention networks (HAN) and (2) knowledge encoding (KE) to effectively capture both explicit and implicit factors of a news article.
- **Evaluation**: Via extensive experiments, we demonstrate that (1) (*accuracy*) KHAN consistently achieves higher accuracies than all competing methods (up to 5.92% higher than the state-of-the-art method), (2) (*efficiency*) KHAN converges within comparable training time/epochs, and (3) (*effectiveness*) each of the main components of KHAN is effective in political stance prediction.

For reproducibility, we have released the code of KHAN and the datasets at https://github.com/yy-ko/khan-www23.

2 RELATED WORK

Language models. The political stance prediction can be seen as a special case of document classification. Thus, general language models [8, 29, 40, 48, 50, 55, 56, 58], which aim at learning to represent words into embedding vectors, could be applied to the political stance prediction problem. Word2Vec [50] is a traditional language model that learns word embeddings based on the similarity between words to preserve local context. However, Word2Vec does not reflect the global context in the entire document [6]. To address the limitation of Word2Vec, GloVe [55] utilizes not only the local relationships among words but also global information of a given document. ELMo [56] aims to learn the meanings of words that can

[1] The details of the user study are described in Appendix A.1.

[2] The data construction details (KG-lib, KG-con, AllSides-L) are provided in Appendix A.2.

be varying depending on the context of a given document, using the pre-trained Bi-LSTM model as contextual information. In addition, *task-agnostic* language models, pre-trained based on large-scale data, can be applied to the political stance prediction problem by fine-tuning them on political news article data. BERT [29], composed of multiple transformer encoders [63], learns the context of given text in a *bidirectional* way with the masked language model (MLM) and the next sentence prediction (NSP). RoBERTa (Robustly optimized BERT) [48], pre-trained on much more training data than BERT, proposes a dynamic masking technique to improve MLM of BERT, achieving higher accuracy than that of BERT. These language models, however, have a limitation in capturing the political characteristics of news articles since they are not designed specially for predicting the political stance of a news article [13, 44, 69].

Political stance prediction models. To overcome the limitation of language models, many researchers have studied methods that consider both text and additional information (e.g., social, linguistic, and political information), which could be useful in political stance prediction [13, 42–44, 61, 69]. HLSTM (Hierarchical LSTM) [42] encodes a news article into an embedding vector using hierarchical LSTM models and it additionally utilizes social context information (e.g., how the news article is spread across users) [33–35] that could reflect the political stance of a news article. Similarly, MAN [44] learns the relationships among words in news articles using multi-head attention networks and considers social and linguistic information necessary to understand the political context of a news article. More recently, knowledge graph (KG) based approaches to political stance prediction have been proposed [13, 69]. A KG-based approach constructs a knowledge graph based on political news articles and uses the political knowledge extracted from the KG as additional information. KGAP (Knowledge Graph Augmented Political perspective detection) [13] represents a given news article as a 4-layer graph (word, sentence, paragraph, and article nodes), and then, it injects the knowledge information to each word node and applies graph neural networks (e.g., R-GCN [59]) to the article graph. KCD (Knowledge walks and textual Cues enhanced political perspective Detection) [69], a state-of-the-art political stance prediction model, generates political knowledge walks via performing random walks in the political KG (like simulating the process of human reasoning) and combines the political knowledge walks with the text of a news article using multi-head attention layers.

3 THE PROPOSED METHOD: KHAN

In this section, we present a novel approach to political stance prediction, Knowledge-aware Hierarchical Attention Networks (**KHAN**). First, we describe the notations and the problem definition. Then, we present two main components of KHAN: hierarchical attention networks (HAN) and knowledge encoding (KE).

3.1 Problem Definition

The notations used in this paper are described in Table 1. KHAN manages two types of embeddings in order to hierarchically learn the relationships among words and the relationships among sentences in a news article: word embeddings ($W = \{w_1, w_2, ..., w_N\}$) and sentence embeddings ($S = \{s_1, s_2, ..., s_l\}$), where w_i (s_i) represents the d-dimensional word (sentence) embedding of the i^{th} word (sentence). In addition to word and sentence embeddings, KHAN also manages external knowledge embeddings to incorporate common and political knowledge in political stance prediction: common and political knowledge embeddings (K^{com}, K^{lib}, and K^{con}).

Table 1: Notations and their descriptions.

Notation	Description
W	a set of word embeddings
S	a set of sentence embeddings
T	the title embedding
w_i, s_j	i^{th} word embedding, j^{th} sentence embedding
d	the embedding dimensionality
N	the total number of words in a dataset
n	the maximum number of words in a sentence
l	the maximum number of sentences in an article
K^{com}	a set of common knowledge embeddings
K^{lib}	a set of liberal knowledge embeddings
K^{con}	a set of conservative knowledge embeddings
α, β	common and political knowledge factors
A, a	news article dataset and a news article
X	a set of learnable parameters
$F(\cdot)$	loss function (i.e., cross-entropy loss)
η	user-defined learning rate

Political stance prediction. This work aims to solve the *political stance prediction* problem: given a news article a, predict its political stance (e.g., [1, 5], where 1 indicates 'left' and 5 indicates 'right'). This problem, a typical supervised learning task, can be defined as follows from the optimization perspective:

$$\min_{w \in \mathbb{R}} \frac{1}{|A|} \sum_{a \in A} F(X, a) + ||X||^2, \tag{1}$$

where X is the set of learnable parameters, A is a given news article dataset, and $F(X, a)$ is the loss function of the parameters X given a news article a. As the loss function, $F(\cdot)$, we adopt cross-entropy loss with the L_2-regularization term.

Based on this problem definition, we optimize the learnable parameters X in an *end-to-end* way. More specifically, to solve the problem represented in Eq 1, we consider SGD as an optimization algorithm. Let X_t be the model parameters at iteration t. Then, X_t is optimized iteratively by the following rule: $X_{t+1} = X_t - \eta \cdot \nabla F(X_t, a)$, where η is the user-defined learning rate.

3.2 Hierarchical Attention Networks (HAN)

As shown in Table 5, the context and tone of a news article are important factors in predicting the political stance of a news article. These factors, however, do not appear in the article explicitly; instead, they are *implicitly* reflected in the overall article. Due to their implicit characteristics, it is very challenging to capture the context and tone of a news article. To address this challenge, in this section, we propose a Hierarchical Attention Networks (**HAN**) that learns (1) the relationships among words, (2) the relationships among sentences, and (3) the relationships between the title and

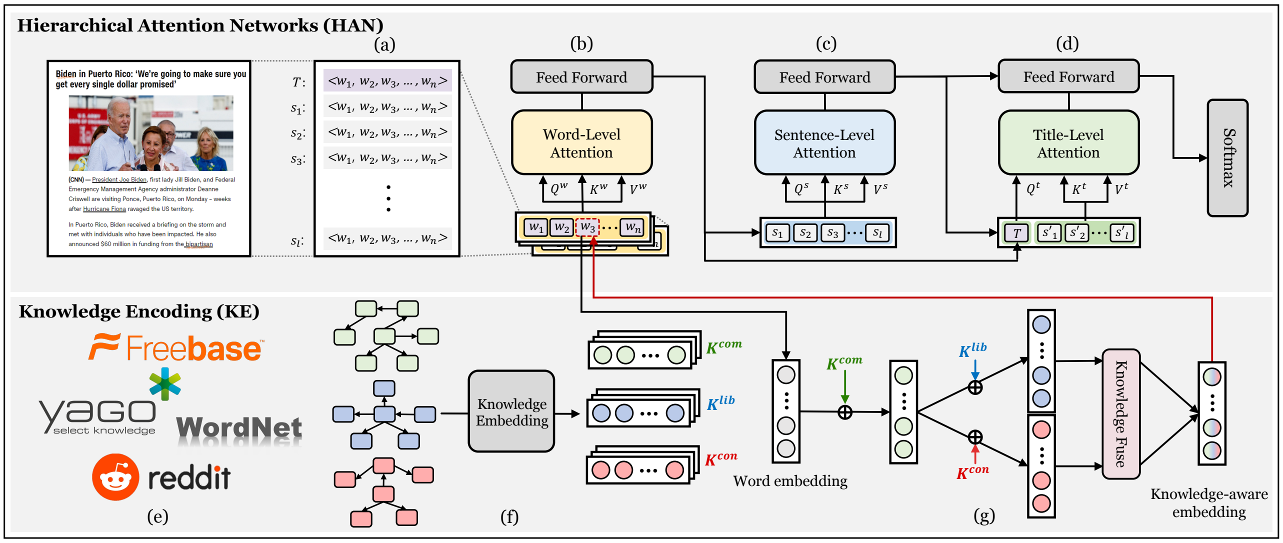

Hierarchical Attention Networks (HAN)
(a)
(b)
(c)
(d)
Biden in Puerto Rico: 'We're going to make sure you get every single dollar promised'
T: $<w_1, w_2, w_3, \ldots, w_n>$
s_1: $<w_1, w_2, w_3, \ldots, w_n>$
s_2: $<w_1, w_2, w_3, \ldots, w_n>$
s_3: $<w_1, w_2, w_3, \ldots, w_n>$
s_l: $<w_1, w_2, w_3, \ldots, w_n>$
Feed Forward
Word-Level Attention
Q^w K^w V^w
Feed Forward
Sentence-Level Attention
Q^s K^s V^s
Feed Forward
Title-Level Attention
Q^t K^t V^t
Softmax
Knowledge Encoding (KE)
Freebase
yago select knowledge
WordNet
reddit
(e)
Knowledge Embedding
(f)
K^{com}
K^{lib}
K^{con}
Word embedding
Knowledge Fuse
Knowledge-aware embedding
(g)

Figure 2: The overview of KHAN: hierarchical attention networks (upper) and knowledge encoding (lower).

sentences (i.e., 3-level hierarchy), for effectively capturing the implicit factors (i.e., context and tone) from news articles. Through the 3-level hierarchy, HAN is able to sequentially capture (1) the local context in each sentence, (2) the global context in an article, and (3) the key idea included in the title.

HAN consists of four layers as illustrated in Figures 2(a)-(d): (a) a pre-processing layer; (b) a word-level attention layer; (c) a sentence-level attention layer; and (d) a title-level attention layer. Now, we describe the four layers and their advantages and challenges.

Pre-processing layer. In general, a news article data is not structured and just represented as a sequence of words (or symbols). Thus, to apply our hierarchical attention networks to a news article, we need to pre-process an input article to represent as a structured form. We pre-process a given news article as follows: Given a news article a, we transform the input article into a set of sentences based on the user-defined separate symbol (e.g., <sep>), and then, represent each sentence as a set of word embedding vectors, $W_j = \{w_1, w_2, ..., w_n\}$, where W_j is the set of word embedding vectors in the j^{th} sentence, w_i is the d-dimensional embedding vector of the i^{th} word, and n is the maximum number of words in a single sentence. Once all news articles in the input dataset are pre-processed, we initialize all word embedding vectors, $\hat{W} = \{w_1, w_2, ..., w_N\}$, where N is the total number of words appearing in the entire dataset.

Word-level attention layer. In the word-level attention layer, we aim to learn 'the relationships among words' in the same sentence, instead of in the entire article, to capture the *local* context of each sentence. Specifically, we apply multi-head self-attention blocks to the set of word embedding vectors in each sentence (i.e., sentence-wise self-attention). Thus, the word-level attention layer for the j^{th} sentence is defined as follows:

$$\tilde{W_j} = MultiHead(Q^w, K^w, V^w), \tag{2}$$

where $Q^w = K^w = V^w = W_j$ and W_j is the set of word embedding vectors in the j^{th} sentence. Then, we pass the output ($\tilde{W_j}$) – the *local-context-aware* word embedding vectors in the j^{th} sentence – to a feed-forward layer, with the same principle of [63], in order to represent the output from the word-level attention layer to better fit the input for the next sentence-level attention layer.

Note that this 'sentence-wise' attention layer of HAN has advantages in terms of both efficiency and effectiveness, compared to the existing 'article-wise' attention methods [5, 12] that take into account all words in the article together. HAN (1) requires the amount of computation much less than the existing methods (*efficient*) and (2) is able to effectively capture the local context without being interfered with the words that are located far away (*effectiveness*) because HAN only learns the relationships among closely located words. However, it is difficult to capture the *global* context of a news article with only the word-level attention layer because this layer is designed specially for capturing the local context of each sentence. To address this limitation, we apply the sentence-level attention layer to the output of this word-level attention layer.

Sentence-level attention layer. The sentence-level attention layer of HAN learns 'the relationships among sentences' in the entire article, to capture the *global* context of the news article. First, the word embedding vectors from the previous layer are averaged in a sentence-wise manner to generate the sentence embedding vectors. Thus, the j^{th} sentence embedding vector (s_j) with d-dimensionality is generated by averaging the d-dimensional word embedding vectors in the j^{th} sentence (i.e., $s_j = Avg(\tilde{W_j})$), where each sentence embedding vector (s_j) has the local context information captured from the previous layer. Then, the sentence-level multi-head self-attention blocks are applied to the set of sentence embedding vectors in each article (i.e., article-wise self-attention). Formally, the sentence-level attention layer for the k^{th} article is defined as the similar way to the word-level attention:

$$\tilde{S_k} = MultiHead(Q^s, K^s, V^s), \tag{3}$$

where $Q^s = K^s = V^s = S_k$ and S_k is the set of the sentence embedding vectors in the k^{th} article. We also pass the *global-context-aware* sentence embedding vectors in k^{th} article ($\tilde{S}_k$) to a feed-forward layer to process the output from this sentence-level attention layer to better fit the next layer (i.e., the title-attention layer).

For the learnable parameters of the word-level and sentence-level attention layers, we use the different sets of parameters in the word-level and the sentence-level attention layers (i.e., $X^{Q^w} \neq X^{Q^s}$) because the related patterns among words (*the local context*) to capture in the word-level layer are highly likely to be different from those among sentences (*the global context*) to capture in the sentence-level layer. As a result, this sentence-level attention layer is able to effectively capture the global context of a news article, based on the previously learned local context, compared to existing non-hierarchical methods counting on all words at once.

Title-level attention layer. The title of a news article, a special sentence having the key idea that the author of the news article hopes to deliver, has been considered as important information in a number of news recommendation systems [17, 31, 65, 67, 70, 73]. Inspired by the importance of the title, we apply the title-level attention layer to the sentence embeddings. The goal of this title-attention layer is (1) to reinforce the key idea included in the title and (2) to filter out relatively unnecessary information in sentences. Let T_k be the title embedding vector of the k^{th} article, then, the title-level attention layer for the k^{th} article is defined as follows:

$$\tilde{S}_k^T = MultiHead(Q^t, K^t, V^t), \tag{4}$$

where $Q^t = T_k$, $K^t = V^t = \tilde{S}_k$, and $\tilde{S}_k$ is the sentence embedding vectors from the previous layer. Thus, by Eq 4, the sentence embedding vectors would be re-weighted in terms of the context of the title (i.e., reinforcing the key idea in the title). Applying the title attention layer in the final stage of HAN, however, might cause the global-context-aware sentence embedding vectors to be biased only to the context of the title too much, thereby leading to the degradation of prediction accuracy. To prevent this problem, we add a residual connection [24, 66] to the output of the title attention layer in order to maintain the global context previously learned from the sentence-level attention layer. Thus, the final output of the title-level attention layer is defined as follows:

$$\tilde{S}_k^* = \tilde{S}_k^T + \tilde{S}_k, \tag{5}$$

where $\tilde{S}_k^*$ is the set of the final sentence embeddings in the k^{th} article. Finally, we aggregate the sentence embeddings, and then pass the aggregated embedding through the output layer.

$$\hat{y} = Predict(a_k), \quad a_k = Avg(\tilde{S}_k^*), \tag{6}$$

where $Predict(\cdot)$ is a softmax layer to predict the political stance of a given news article, $\hat{y}$.

As a result, via the 3-level hierarchical process, HAN is able to effectively capture both the local and global context implicitly reflected in a news article. We will verify the effectiveness of HAN in improving the model accuracy of KHAN in Section 4.

3.3 Knowledge Encoding (KE)

As shown in Appendix A.1, it is crucial to understand the *interpretation* and *sentiment* to real-world entities such as keywords and persons in predicting the political stance of a news article. In general, however, it often occurs that the information about many real-world entities is not clearly provided in news articles. For example, famous politicians (e.g., 'Barack Obama' and 'Donald Trump') are not always directly described in a news article. From this motivation, we propose **K**nowledge **E**ncoding (**KE**) that pre-trains the external knowledge (both common and political) related to the real-world entities and injects the external knowledge into the corresponding words appearing in a news article for accurate political stance prediction. Regarding political knowledge, we take into account two different political knowledge (i.e., liberal and conservative) separately in the knowledge encoding.

Table 2: Statistics of political KGs.

	KGAP [13]	KG-lib	KG-con
# of source posts	-	219,915	276,156
# of entities	1,071	5,581	6,316
# of relations	10,703	29,967	33,207
Political stances	Both	Liberal	Conservative

This approach has advantages in capturing subtle but important difference in knowledge, related to real-world entities, varying depending on political stances, compared to the existing knowledge-based methods [13, 69] that consider only unified political knowledge. However, there is a technical challenge about how to incorporate the three different knowledge (one common and two political knowledge) into the process of the political stance prediction. To this challenge, we propose a simple but effective algorithm for knowledge injection, where KE learns how to fuse the three types of knowledge. Figures 2(e)-(g) illustrate the three stages of KE: (1) knowledge preparation; (2) embedding; and (3) injection.

Knowledge preparation. First, we prepare both common and political knowledge for real-world entities appearing in news articles. In terms of common knowledge, there are many well-designed knowledge graphs (KG) such as YAGO [54], Freebase [3], and WordNet [51] built from large-scale real-world datasets, where a node represents an entity and an edge represents a relation between two entities. Among them, we choose YAGO [54] as the source of common knowledge in this work because YAGO consists of general knowledge about real-world entities (e.g., people, cities, countries, movies, and organizations). On the other hand, political knowledge has been rarely studied except for some works [13, 69] that built a single political KG based on the political entities and their relations from U.S. political websites (e.g., AFL-CIO and Heritage Action) Unfortunately, it is challenging to accurately represent the relations among political entities in a single knowledge graph because *the interpretation and sentiment to political entities and their relations can be different depending on the political stance.* To address this challenge, we (1) collected 496,071 political-related posts from the U.S. political community websites[3], (2) extracted 18 political entities, using an NER (named entity recognition) method [64], and (3) constructed two different political knowledge graphs: KG-lib and KG-con, where each data point is represented as a triplet: <head entity, relation, tail entity>. Table 2 shows the statistics of the political knowledge graphs. The details about the knowledge graph construction are included in Appendix A.2.

[3] https://www.reddit.com/r/Liberal/, https://www.reddit.com/r/Conservative/

Algorithm 1 Knowledge injection of KE

```
Require: news article $a$, word embeddings $W$, three types of
    knowledge embeddings: $K^{com}$, $K^{lib}$, $K^{con}$, knowledge control
    factors $\alpha, \beta$
1: Initialize $W^* \leftarrow \phi$
2: for word $i \in a$ do
3:     $e \leftarrow W[i]$
4:     $e^{com} \leftarrow (1-\alpha) \cdot e \oplus \alpha \cdot K^{com}[i]$
5:     $e^{lib} \leftarrow (1-\beta) \cdot e^{com} \oplus \beta \cdot K^{lib}[i]$
6:     $e^{con} \leftarrow (1-\beta) \cdot e^{com} \oplus \beta \cdot K^{con}[i]$
7:     $W^*[i] \leftarrow \mathsf{Fuse}([e^{lib} \| e^{con}]) \oplus e$
8: end for
9: return $W^*$
```

Knowledge embedding. Next, we apply knowledge embedding methods [4, 11, 41, 62, 71], aiming to learn the relations among entities, to the three KGs independently, in order to represent three types of embedding vectors for each entity: K^{com}, K^{lib}, and K^{con} (i.e., common, liberal, and conservative). Since this work is *agnostic* to the knowledge embedding method, any knowledge embedding methods could be applied to KE. As the knowledge embedding method, we consider three recent methods: RotatE [62], ModE [71], and HAKE [71]. In this paper, we omit the details of the knowledge embedding methods because it is beyond the scope of this work, but we include the knowledge graph completion accuracies of the three knowledge embedding methods in Appendix A.3.

Knowledge injection. As we mentioned in the beginning of Section 3.3, there is a technical challenge about how to incorporate the three different knowledge into the process of predicting the political stance of a news article. Specifically, it is difficult to determine (1) how to fuse the three types of knowledge and (2) how much amount of each knowledge is needed, for accurate political stance prediction. To this challenge, we propose a simple but effective knowledge injection algorithm that fuses the three types of knowledge for real-world entities and injects them into the corresponding words appearing in a news article.

The process of the knowledge injection of KE is illustrated in Figure 2(g) and described in Algorithm 1. For each word in a given article, KE first injects the common knowledge of the word ($K^{com}[i]$) into the corresponding word embedding (lines 3-4 in Algorithm 1), where α is the common knowledge control factor and '$\oplus$' means the element-wise addition. Then, we add the political knowledge embeddings to the common knowledge injected embedding e^{com} with the political knowledge control factor β independently (lines 5-6). The two embeddings with different political knowledge, e^{lib} and e^{con}, are concatenated and passed through a fully-connected layer, $\mathsf{Fuse}(\cdot)$ that fuses the two embeddings into a single knowledge-aware embedding (line 7), which plays a role to learn how to fuse the different political knowledge for accurate prediction. We also add the original word embedding to the knowledge-aware embedding (i.e., residual connection). As a result, the knowledge-aware word embeddings can be used across the hierarchical news encoding process of HAN. We will verify the effectiveness of KE and the impacts of its hyperparameters α and β in improving the political stance prediction accuracy of KHAN in Section 4.

Table 3: Statistics of political news article datasets.

Dataset	# of articles	Class distribution
SemEval	645	407 / 238
AllSides-S	14.7k	6.6k / 4.6k / 3.5k
AllSides-L	719.2k	112.4k / 202.9k / 99.6k / 62.6k / 241.5k

4 EXPERIMENTAL VALIDATION

In this section, we comprehensively evaluate KHAN by answering the following evaluation questions:

- EQ1. To what extent does KHAN improve existing methods in terms of the model accuracy in the political stance prediction?
- EQ2. How fast does KHAN converge in terms of time and epochs?
- EQ3. How effective are the main components of KHAN in terms of improving the model accuracy in political stance prediction?
- EQ4. How sensitive is the model accuracy of KHAN to the hyperparameters α and β?

4.1 Experimental Setup

Datasets. We evaluate KHAN with three real-world news article datasets, SemEval [30], AllSides-S [42], and AllSides-L. Table 3 shows the statistics of the news article datasets. SemEval consists of 645 articles with 2 classes (hyperpartisan and center) and AllSides-S consists of 14,783 articles with 3 classes (left, center, and right). For training, we use 10-fold and 3-fold cross validations for SemEval and AllSides-S, respectively, as the same in previous works [13, 42, 44, 69]. For extensive evaluation, we construct a large-scale political news dataset, AllSides-L with 719,256 articles with 5 classes (left, lean left, center, lean right, and right)[4]. We split the AllSides-L dataset into training and validation sets: 647,330 articles in the training set and 71,926 articles in the validation set.

Baseline methods. We compare KHAN with seven baseline methods: five text-based methods [29, 48, 50, 55, 56] and two knowledge-based political stance prediction methods [13, 69]. Word2Vec [50], GloVe [55], and ELMo [56] are general language models that aim to capture context from text. We also use pre-trained BERT [29] and RoBERTa [48] by fine-tuning on training datasets. KGAP [13] is a knowledge-aware approach that leverages a political knowledge graph with graph neural networks (e.g., R-GCN [59]). KCD [69], the state-of-the-art model, considers knowledge walks from a political knowledge graph and textual cues in political stance prediction.

Implementation details. We use PyTorch 1.10.0 to implement all methods including KHAN on Ubuntu 20.04 OS. We run our experiments on the machine equipped with an Intel i7-9700k CPU with 64 GB memory and a NVIDIA RTX 2080 Ti GPU, installed with CUDA 11.3 and cuDNN 8.2.1. We set the batch size b as 16 for all datasets as the same in the previous works [13, 69]. We use the Adam optimizer [32] with the learning rate η = 1e-3 and the weight decay factor 5e-2 for all datasets. As a learning rate scheduler [21, 27, 36], we use the 'ReduceLROnPlateau' scheduler of PyTorch that reduces the learning rate η by 1/2 when the training loss has stopped improving for a 'patience' number of epochs in a row (we set the patience epoch as 5).

[4] The data construction details for AllSides-L are provided in Appendix A.2.

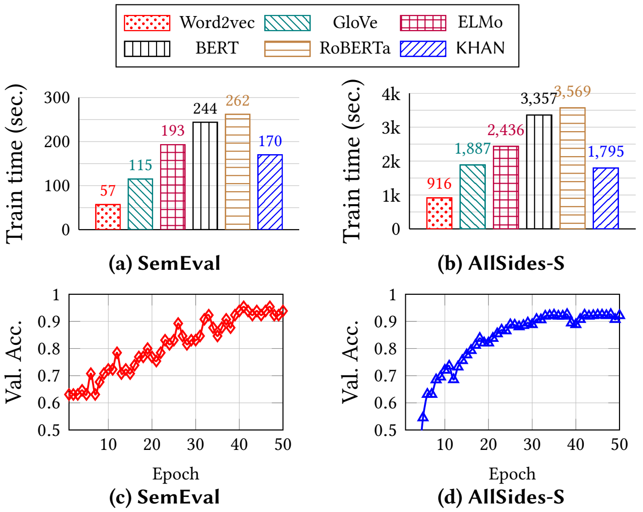
Table 4: Comparison of the model accuracy on three real-world datasets (The bold font indicates the best results).

Method	Dataset		
	SemEval	**AllSides-S**	**AllSides-L**
Word2Vec [50]	0.7027	0.4858	0.4851
GloVe [55]	0.8071	0.7101	0.6354
ELMo [56]	0.8678	0.8197	0.7483
BERT [29]	0.8692	0.8246	0.7812
RoBERTa [48]	0.8708	0.8535	0.8222
KGAP [13]	0.8956	0.8602	N/A
KCD [69]	0.9087	0.8738	N/A
KHAN-RotatE	0.9426	0.9151	0.8584
KHAN-HAKE	0.9395	0.9216	0.8563
KHAN-ModE	**0.9521**	**0.9256**	**0.8617**

4.2 Experimental Results

EQs 1-2. Accuracy and Efficiency. In this experiment, we evaluate the accuracy and efficiency of the proposed KHAN in the political stance prediction. We train KHAN on the three real-world political news article datasets (50 epochs) with varying knowledge embedding methods (RotatE, HAKE, and ModE). For the SemEval and AllSides-L datasets, we apply k-fold cross validations (10-fold for SemEval and 3-fold for AllSides-S). Due to the page limit, we report the averaged accuracy of k-fold cross validations and include the entire results with the standard deviation in Appendix A.3. For AllSides-L, we train KHAN on the training set (647k articles) and measure the model accuracy using the validation set (71.9k articles), which is not used in the training process.

As shown in Table 4[5], KHAN consistently outperforms all baseline methods in terms of the model accuracy, regardless of knowledge embedding methods. Specifically, KHAN improves the state-of-the-art method, KCD [69] by 4.77% and 5.92% in SemEval and AllSides-S datasets, respectively. These improvements over KCD are significant, given that KCD has already achieved quite high accuracies in those datasets. We also evaluate KHAN on a large-scale dataset (AllSides-L) which is 48× larger and has more classes (i.e., more difficult to predict) than the previous largest one (AllSides-S). KHAN still significantly outperforms all baseline methods although the accuracy of KHAN decreases to some extent in AllSides-L, compared with the other datasets. In addition, we have conducted the t-tests with a 95% confidence level and verified that the improvement of KHAN over all baseline methods are statistically significant (i.e., the p-values are below 0.05). As a result, KHAN consistently outperforms state-of-the-art methods in all datasets, including a our own dataset (AllSides-L), with a 95% confidence level, which implies that KHAN has a good generalizability. In addition to generalizability, in terms of applicability, KHAN could be applied to other languages since KHAN does not depend on any linguistic features (e.g., word-order and grammatical features). As an example, we have successfully applied KHAN to a web-based platform for diverse political news consumption in non-English.



(a) SemEval **(b) AllSides-S**

(c) SemEval **(d) AllSides-S**

Figure 3: The training time and convergence rate with respect to training epochs of KHAN on SemEval and AllSides-S.

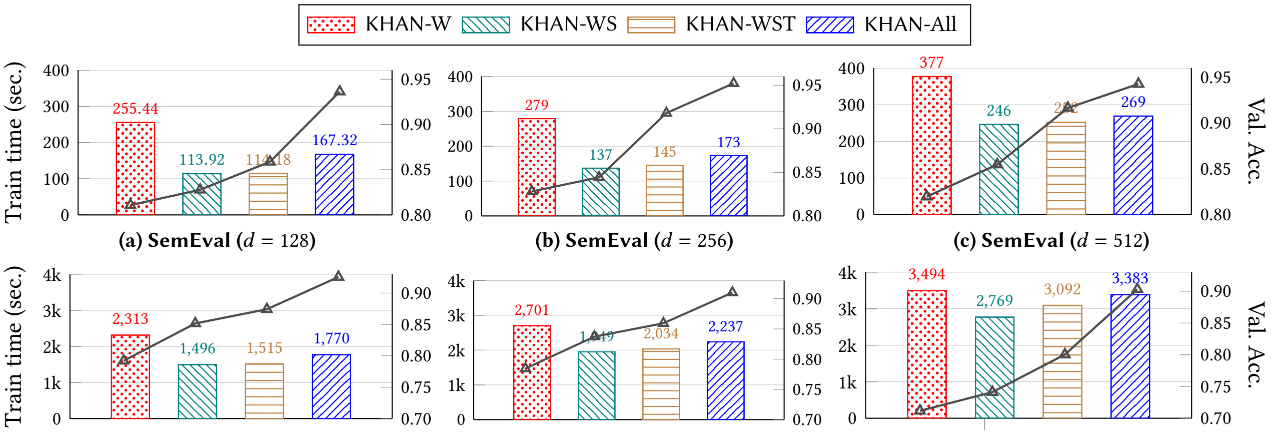
We also evaluate KHAN in terms of the training time and convergence rate with respect to training epoch. Figures 3(a)-(b) represent the training time for 50 epochs of each method and Figures 3(c)-(d) show the validation accuracy of KHAN with respect to the training epoch. KHAN finishes its training in shorter time than ELMo, BERT, and RoBERTa (even faster than GloVe in AllSides-S), while achieving the highest accuracy at the same time. Considering the low accuracies of Word2Vec and GloVe (at least 15% less than KHAN), these results verify the efficiency of KHAN. In terms of convergence rate, KHAN converges to high accuracy within only around 40 epochs (Figure 3(c)-(d)). All in all, these results demonstrate that KHAN is able to *effectively* and *efficiently* solve the political stance prediction problem by employing the proposed HAN and KE.

EQ3. Ablation Study. In this experiment, we verify the effectiveness of the HAN and KE. We compare the four versions of KHAN:

- KHAN-W: a baseline with only the word-level attention layer.
- KHAN-WS: the version with the word and sentence layers.
- KHAN-WST: the version with HAN (word, sentence, and title).
- KHAN-All: the original version with both HAN and KE.

We train each version of KHAN with varying the embedding dimensionality (d = 128, 256, and 512) on SemEval (10-fold) and AllSides-S (3-fold), and measure the accuracy and training time. As shown in Figure 4, KHAN-WS consistently improves KHAN-W in both the accuracy and training performance. This result demonstrates that our hierarchical approach not only (1) (*efficiency*) requires much less computation overhead than non-hierarchical existing methods, but also (2) (*effectiveness*) captures the context implicitly reflected in a news article successfully, as we claimed in Section 3.2. KHAN-WST further improves the model accuracy with only a very small amount of overhead, thereby verifying the importance of the title that has the key idea of the news article in political stance prediction. Finally, KHAN-All always achieves the highest accuracy in all cases, regardless of datasets and embedding dimensionality. This result validates that external knowledge for real-world entities appearing in a news article is indeed important in predicting its political stance, as we claimed in Section 3.3.

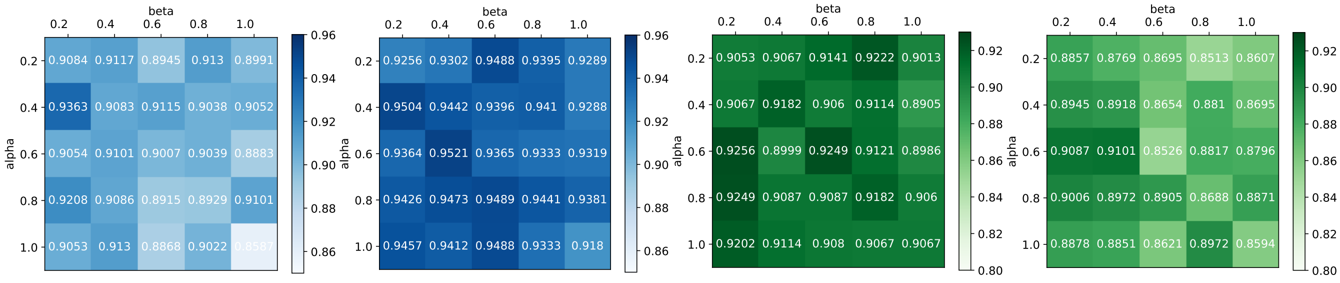
[5] For SemEval and AllSides-S, we use the results for baseline methods reported in [69]. We also obtain the results for baseline methods and include them in Appendix A.3.



(a) SemEval (d = 128) (b) SemEval (d = 256) (c) SemEval (d = 512)

(d) AllSides-S (d = 128) (e) AllSides-S (d = 256) (f) AllSides-S (d = 512)

Figure 4: Effectiveness of the main components of KHAN in terms of the training time (bar) and model accuracy (line).



alpha \ beta	0.2	0.4	0.6	0.8	1.0
0.2	0.9084	0.9117	0.8945	0.913	0.8991
0.4	0.9363	0.9083	0.9115	0.9038	0.9052
0.6	0.9054	0.9101	0.9007	0.9039	0.8883
0.8	0.9208	0.9086	0.8915	0.8929	0.9101
1.0	0.9053	0.913	0.8868	0.9022	0.8587

(a) SemEval (d = 128)

alpha \ beta	0.2	0.4	0.6	0.8	1.0
0.2	0.9256	0.9302	0.9488	0.9395	0.9289
0.4	0.9504	0.9442	0.9396	0.941	0.9288
0.6	0.9364	0.9521	0.9365	0.9333	0.9319
0.8	0.9426	0.9473	0.9489	0.9441	0.9381
1.0	0.9457	0.9412	0.9488	0.9333	0.918

(b) SemEval (d = 256)

alpha \ beta	0.2	0.4	0.6	0.8	1.0
0.2	0.9053	0.9067	0.9141	0.9222	0.9013
0.4	0.9067	0.9182	0.906	0.9114	0.8905
0.6	0.9256	0.8999	0.9249	0.9121	0.8986
0.8	0.9249	0.9087	0.9087	0.9182	0.906
1.0	0.9202	0.9114	0.908	0.9067	0.9067

(c) AllSides-S (d = 128)

alpha \ beta	0.2	0.4	0.6	0.8	1.0
0.2	0.8857	0.8769	0.8695	0.8513	0.8607
0.4	0.8945	0.8918	0.8654	0.881	0.8695
0.6	0.9087	0.9101	0.8526	0.8817	0.8796
0.8	0.9006	0.8972	0.8905	0.8688	0.8871
1.0	0.8878	0.8851	0.8621	0.8972	0.8594

(d) AllSides-S (d = 256)

Figure 5: The impact of hyperparameters α and β on the model accuracy of KHAN in political stance prediction.

EQ4. Hyperparameter sensitivity. Finally, we evaluate the impacts of the hyperparameters α and β on the model accuracy of KHAN. As explained in Section 3.3, the hyperparameter α (β) controls the amount of common (political) knowledge is injected to the corresponding words in a given news article. Thus, as α (β) becomes lower, the more amount of the common (political) knowledge is injected to the entities appearing in a new article. While, as α (β) becomes larger, the less amount of the common (political) knowledge is injected. For an extreme case, if α (β) is 1, the common (political) knowledge is not used. We measure the model accuracy with varying α and β from 0.2 to 1.0 on SemEval and AllSides-S.

As clearly illustrated in Figure 5, KHAN with lower β tends to achieve higher accuracy than KHAN with larger β (i.e., the left-hand side of each heatmap tends to be darker than its right-hand side in each row). On the other hand, KHAN with larger α and β shows relatively poor performance in political stance prediction accuracy (i.e, the bottom right side of each heatmap tends to be brighter than other sides). Specifically, KHAN with $\alpha = 1$ and $\beta = 1$ (i.e., both common and political knowledge not used) shows the worst results in the SemEval dataset. These results verify that external knowledge for real-world entities is indeed useful in predicting the political stance of a news article. Note that KHAN with α and β below 0.6 consistently outperforms all baseline methods. Based on these results, we believe that the model accuracy of KHAN is not sensitive to the hyperparameters α and β, and we recommend to set the hyperparameters α and β as below 0.6.

5 CONCLUSION

In this paper, via a carefully-designed user study, we observe that both explicit and implicit factors (i.e., the context/tone and external knowledge for real-world entities) are important in predicting the political stances of news articles. Based on the observations, we propose a novel approach to accurate political stance prediction, KHAN that successfully captures the local and global context of a new article with the two key components: (1) hierarchical attention networks (HAN) to learn the relationships among words, sentences, and the title in a news article with the 3-level hierarchy and (2) knowledge encoding (KE) to incorporate the three types of useful knowledge for real-world entities into the process of the political stance prediction. Via the extensive experiments, we demonstrate that (1) (*accuracy*) KHAN consistently achieves higher accuracies than all baseline methods, (2) (*efficiency*) KHAN is able to converge to high accuracies within comparable training time (epochs), and (3) (*effectiveness*) the key components of KHAN (HAN and KE) are quite effective in improving the model accuracy of KHAN.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) (2018R1A5A7059549), and Institute of Information & Communications Technology Planning & Evaluation (IITP) (RS-2022-00155586, 2022-0-00352, 2020-0-01373). Hanghang Tong was partially supported by NSF (1947135, 2134079, and 1939725).

REFERENCES

[1] Christopher A Bail, Lisa P Argyle, Taylor W Brown, John P Bumpus, Haohan Chen, MB Fallin Hunzaker, Jaemin Lee, Marcus Mann, Friedolin Merhout, and Alexander Volfovsky. 2018. Exposure to opposing views on social media can increase political polarization. *In Proceedings of the National Academy of Sciences* 115, 37 (2018), 9216–9221.

[2] Eytan Bakshy, Solomon Messing, and Lada A Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* 348, 6239 (2015), 1130–1132.

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *In Proceedings of the ACM SIGMOD international conference on Management of data (SIGMOD)*. 1247–1250.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[5] Junyi Chen, Lan Du, Ming Liu, and Xiabing Zhou. 2022. Mulan: A Multiple Residual Article-Wise Attention Network for Legal Judgment Prediction. *Transactions on Asian and Low-Resource Language Information Processing* 21, 4 (2022), 1–15.

[6] Minmin Chen. 2017. Efficient Vector Representation for Documents through Corruption. In *In Proceedings of International Conference on Learning Representations(Poster)*.

[7] Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. 2021. The echo chamber effect on social media. *In Proceedings of the National Academy of Sciences* 118, 9 (2021), e2023301118.

[8] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *In Proceedings of International Conference on Learning Representations*.

[9] Wesley Cota, Silvio C Ferreira, Romualdo Pastor-Satorras, and Michele Starnini. 2019. Quantifying echo chamber effects in information spreading over political communication networks. *EPJ Data Science* 8, 1 (2019), 1–13.

[10] Lincoln Dahlberg. 2001. The Internet and democratic discourse: Exploring the prospects of online deliberative forums extending the public sphere. *Information, communication & society* 4, 4 (2001), 615–633.

[11] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *In Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[12] Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. International Joint Conferences on Artificial Intelligence.

[13] Shangbin Feng, Zilong Chen, Wenqian Zhang, Qingyao Li, Qinghua Zheng, Xiaojun Chang, and Minnan Luo. 2021. KGAP: Knowledge Graph Augmented Political Perspective Detection in News Media. *arXiv preprint arXiv:2108.03861* (2021).

[14] Seth Flaxman, Sharad Goel, and Justin M Rao. 2016. Filter bubbles, echo chambers, and online news consumption. *Public opinion quarterly* 80, S1 (2016), 298–320.

[15] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *In Proceedings of the 2018 world wide web conference*. 913–922.

[16] R Kelly Garrett and Paul Resnick. 2011. Resisting political fragmentation on the Internet. *Daedalus* 140, 4 (2011), 108–120.

[17] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *In Proceedings of The Web Conference 2020*. 2863–2869.

[18] Matthew Gentzkow and Jesse M Shapiro. 2010. What drives media slant? Evidence from US daily newspapers. *Econometrica* 78, 1 (2010), 35–71.

[19] Nabeel Gillani, Ann Yuan, Martin Saveski, Soroush Vosoughi, and Deb Roy. 2018. Me, my echo chamber, and I: introspection on social media polarization. In *In Proceedings of the 2018 World Wide Web Conference*. 823–831.

[20] Swapna Gottopati, Minghui Qiu, Yanchuan Sim, Jing Jiang, and Noah Smith. 2013. Learning topics and positions from debatepedia. Association for Computational Linguistics.

[21] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, Large Minibatch SGD: Training Imagenet in 1 Hour. *arXiv preprint arXiv:1706.02677* (2017).

[22] Soeun Han, Yunyong Ko, Yushim Kim, Seong Soo Oh, Heejin Park, and Sang-Wook Kim. 2022. D-FEND: A Diffusion-based Fake News Detection Framework for News Articles Related to COVID-19. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (ACM SAC)*. 1771–1778.

[23] Taha Hassan. 2019. Trust and trustworthiness in social recommender systems. In *Companion Proceedings of The 2019 World Wide Web Conference*. 529–532.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[25] Judith Irvine, Bambi Schieffelin, CM Series, Marjorie Harness Goodwin, Joel Kuipers, Don Kulick, John Lucy, Elinor Ochs, et al. 1992. *Rethinking context: Language as an interactive phenomenon*. Number 11. Cambridge University Press.

[26] Youngseung Jeon, Bogoan Kim, Aiping Xiong, Dongwon Lee, and Kyungsik Han. 2021. ChamberBreaker: Mitigating the Echo Chamber Effect and Supporting Information Hygiene through a Gamified Inoculation System. *In Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–26.

[27] Tyler Johnson, Pulkit Agrawal, Haijie Gu, and Carlos Guestrin. 2020. AdaScale SGD: A User-Friendly Algorithm for Distributed Training. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 4911–4920.

[28] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems–Survey and roads ahead. *Information Processing & Management* 54, 6 (2018), 1203–1227.

[29] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *In Proceedings of NAACL-HLT*. 4171–4186.

[30] Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *In Proceedings of the 13th International Workshop on Semantic Evaluation*. 829–839.

[31] Taeho Kim, Yungi Kim, Yeon-Chang Lee, Won-Yong Shin, and Sang-Wook Kim. 2022. Is It Enough Just Looking at the Title? Leveraging Body Text To Enrich Title Words Towards Accurate News Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*. 4138–4142.

[32] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *In Proceedings of International Conference on Learning Representations(Poster)*.

[33] Yunyong Ko, Dong-Kyu Chae, and Sang-Wook Kim. 2016. Accurate path-based methods for influence maximization in social networks. In *Proceedings of the ACM Web Conference (WWW)*. 59–60.

[34] Yunyong Ko, Dong-Kyu Chae, and Sang-Wook Kim. 2018. Influence maximisation in social networks: A target-oriented estimation. *Journal of Information Science* 44, 5 (2018), 671–682.

[35] Yunyong Ko, Kyung-Jae Cho, and Sang-Wook Kim. 2018. Efficient and effective influence maximization in social networks: a hybrid-approach. *Information Sciences* 465 (2018), 144–161.

[36] Yunyong Ko, Dongwon Lee, and Sang-Wook Kim. 2022. Not All Layers Are Equal: A Layer-Wise Adaptive Approach Toward Large-Scale DNN Training. In *Proceedings of the ACM Web Conference (WWW)*. 1851–1859.

[37] Travis Kriplean, Jonathan Morgan, Deen Freelon, Alan Borning, and Lance Bennett. 2012. Supporting reflective public thought with considerit. In *In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. 265–274.

[38] Arie W Kruglanski. 1996. Motivated social cognition: Principles of the interface. (1996).

[39] Margot Kuttschreuter, Jan Martien Gutteling, and Maureen De Hond. 2011. Framing and tone-of-voice of disaster media coverage: The aftermath of the Enschede fireworks disaster in the Netherlands. *Health, risk & society* 13, 3 (2011), 201–220.

[40] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *In Proceedings of International Conference on Learning Representations*.

[41] Yeon-Chang Lee, JaeHyun Lee, Dongwon Lee, and Sang-Wook Kim. 2022. THOR: Self-Supervised Temporal Knowledge Graph Embedding via Three-Tower Graph Convolutional Networks. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. 1035–1040.

[42] Chang Li and Dan Goldwasser. 2019. Encoding social information with graph convolutional networks forpolitical perspective detection in news media. In *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2594–2604.

[43] Chang Li and Dan Goldwasser. 2021. Mean: Multi-head entity aware attention networkfor political perspective detection in news media. In *In Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics, 66–75.

[44] Chang Li and Dan Goldwasser. 2021. Using social and linguistic information to adapt pretrained representations for political perspective identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 4569–4579.

[45] Q Vera Liao and Wai-Tat Fu. 2014. Can you hear me now? Mitigating the echo chamber effect by source position indicators. In *In Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 184–196.

[46] Q Vera Liao, Wai-Tat Fu, and Sri Shilpa Mamidi. 2015. It is all about perspective: An exploration of mitigating selective exposure with aspect indicators. In *In Proceedings of the 33rd annual ACM conference on Human factors in computing systems*. 1439–1448.

[47] Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander G Hauptmann. 2006. Which side are you on? Identifying perspectives at the document and sentence levels. In *In Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*. 109–116.

[48] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[49] Gabriel Machado Lunardi. 2019. Representing the filter bubble: Towards a model to diversification in news. In *In Proceedings of the 19th International Conference on Conceptual Modeling*. Springer, 239–246.

[50] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[51] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[52] Sean Munson, Stephanie Lee, and Paul Resnick. 2013. Encouraging reading of diverse political viewpoints with a browser widget. In *In Proceedings of The International AAAI Conference on Web and Social Media*, Vol. 7. 419–428.

[53] Matti Nelimarkka, Jean Philippe Rancy, Jennifer Grygiel, and Bryan Semaan. 2019. (Re) Design to Mitigate Political Polarization: Reflecting Habermas' ideal communication space in the United States of America and Finland. *In Proceedings of the ACM on Human-computer Interaction* 3, CSCW (2019), 1–25.

[54] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *In Proceedings of the European Semantic Web Conference*. Springer, 583–596.

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

[56] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[57] Walter Quattrociocchi, Antonio Scala, and Cass R Sunstein. 2016. Echo chambers on Facebook. *Available at SSRN 2795110* (2016).

[58] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[59] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.

[60] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.

[61] Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance detection with hierarchical attention network. In *In Proceedings of the 27th International Conference on Computational Linguistics*. 2399–2409.

[62] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *In Proceedings of International Conference on Learning Representations*.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[64] Sergey Vychegzhanin and Evgeny Kotelnikov. 2019. Comparison of named entity recognition tools applied to news articles. In *2019 Ivannikov Ispras Open Conference (ISPRAS)*. IEEE, 72–77.

[65] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *In Proceedings of the 2018 world wide web conference*. 1835–1844.

[66] Hong Wei, Hao Zhou, Jangan Sankaranarayanan, Sudipta Sengupta, and Hanan Samet. 2018. Residual convolutional lstm for tweet count prediction. In *Companion Proceedings of the Web Conference 2018 (TheWebConf)*. 1309–1316.

[67] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: neural news recommendation with personalized attention. In *In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2576–2584.

[68] Savvas Zannettou, Barry Bradlyn, Emiliano De Cristofaro, Haewoon Kwak, Michael Sirivianos, Gianluca Stringini, and Jeremy Blackburn. 2018. What is gab: A bastion of free speech or an alt-right echo chamber. In *Companion Proceedings of the The Web Conference 2018*. 1007–1014.

[69] Wenqian Zhang, Shangbin Feng, Zilong Chen, Zhenyu Lei, Jundong Li, and Minnan Luo. 2022. KCD: Knowledge Walks and Textual Cues Enhanced Political Perspective Detection in News Media. In *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*. Association for Computational Linguistics, 4129–4140.

[70] Xuanyu Zhang, Qing Yang, and Dongliang Xu. 2021. Combining explicit entity graph with implicit text information for news recommendation. In *Companion Proceedings of the Web Conference 2021*. 412–416.

[71] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3065–3072.

[72] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *In Proceedings of the Web Conference 2021*. 401–412.

[73] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5973–5980.

[74] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *In Proceedings of the 14th international conference on World Wide Web*. 22–32.

A APPENDIX

In this appendix, we describe the detailed information of our user study (Appendix A.1), data construction (Appendix A.2), and the additional experimental results about the reliability of our experiments about political stance prediction (Appendix A.3).

A.1 User Study

In this section, we describe the details of our user study to investigate the important factors that real-world users take into account to determine the political stance of a news articles.

Setup. We have recruited 136 respondents in total from Amazon Mechanical Turk[6]. For a fair and reliable user study, we chose the respondents considering a variety of aspects such as gender (male/female), age (from below 20 to above 50), education (from high school or less, to university graduation and above), ethnicity (e.g., Caucasian, African American, Asian, Hispanic/Latino, and Others) and political stance (very liberal, somewhat liberal, neutral, somewhat conservative, very conservative). We selected six news articles with different political stances and thirteen political-related factors. Regarding the topic of a news article, we considered three different news topics which are highly related to political stances (i.e., Health, Environment, and Tax)[7]. For political-related factors, we carefully chose thirteen factors in total, which have been studied in [18, 20, 25, 39, 47], such as context/tone, keywords, person names, topic/issue, title, images of a news article, slang words used in an article, and social/religious factors.

User study protocol. Then, we (1) provided the six news articles (the title, body, and image) and the thirteen carefully-chosen factors to the respondents, (2) asked them to respond how important each factor is in their decision with a scale [1:(not at all) - 5:(very much)], and (3) assessed each factor for political stance identification by averaging the scores rated by the respondents.

Table 5: Result of the user study: the top-5 factors in political stance predictions and their importance scores.

Rank	Factor name	Importance score (1-5)
1	Context	4.19 ± 0.94
2	Keyword	4.01 ± 0.88
3	Person	3.94 ± 0.96
4	Tone	3.93 ± 1.13
5	Freq. used word	3.35 ± 1.07

Result and analysis. Table 5 shows the top-5 important factors in political stance prediction and their scores. This result shows that the "context" of a new article is the most important factor in deciding its political stance, followed by keyword, person, tone, and frequently used word. This user study result implies that it is crucial (1) to learn the relationships among words/sentences to capture the context and tone of a news article, which is *implicitly* reflected in a news article, and (2) to understand the interpretation and sentiment to real-world entities (e.g., keyword and person), which *explicitly* appear in a news article.

A.2 Data Construction

In this section, we describe the process of data construction for our datasets: (1) a large-scale political news datasets (AllSides-L) and (2) two political knowledge graphs (KG-lib and KG-con). All the datasets are available at: https://github.com/yy-ko/khan-www23.

AllSides-L. As explained in 4.1, we constructed a large-scale political news dataset, AllSides-L. We collected 719,256 articles with 5 classes (left, lean left, center, lean right, and right) from Allsides.com [8], which is an American website to alleviate side effects by media bias and misinformation. Allsides.com provides political-related news articles with diverse political stances, where it decides the political class (e.g., left or right) of each news article based on its news outlet (e.g., CNN or Fox). For the political label decision, Allsides.com uses the three-step process: Each news outlet is labeled by (1) domain experts, (2) user studies by average people with diverse political stances, and (3) majority voting by others who do not engage in the user studies. Thanks to its careful labeling, the political stances of news articles by Allsides.com are generally used as the ground truth [13, 69]. For the more reliability of AllSides-L, we consider only the news articles from the outlets receiving high scores (7-out-of-8 or better) on their labels at majority voting.

KG-lib and KG-con. We constructed two different political knowledge graphs (KGs), KG-lib and KG-con, via a three-step process: (1) data collection, (2) entity/relation extraction, and (3) data cleansing. We first collected 219,915 posts and 276,156 posts from the U.S. liberal and conservative communities, respectively (496,071 posts in total). Since the raw posts could include many political-unrelated entities, we need to extract political entities and their relations from the raw posts. To this end, we extracted 18 political-related entities and their relations, using a state-of-the-art NER (Named Entity Recognition) method [64]. Via this step, each data point is represented as a triplet: <head entity, relation, tail entity>. For more reliability of the political knowledge graphs, we manually remove noises from the extracted triplets. Finally, we constructed the two political knowledge graphs, KG-lib (5,581 entities and 29,967 relations) and KG-con (6,316 entities and 33,207 relations).

Quality of the political knowledge graphs. We also evaluate the quality of our political knowledge graphs. As explained in Section 3.3, we use three recent knowledge embedding methods to KE: RotatE [62], ModE [71], and HAKE [71]. We apply each knowledge embedding method to the two political knowledge graphs (i.e., KG-lib and KG-con) with varying the embedding dimensionality ($d = 128, 256, 512$), and measure the quality of knowledge graphs by using five knowledge graph completion metrics: MR (mean rank), MMR (mean reciprocal rank), HITS@1, HITS@3, and HITS@10. Tables 6, 7, and 8 show the results. The quality of the political knowledge embedding tends to be improved as the dimensionality of embedding increases. Note that the quality of political knowledge embeddings could be improved in two aspects: (1) extending the scale of political knowledge graphs (KG) and (2) developing a new knowledge embedding method specialized in the political stance prediction. In future work, we plan to extend the scale of political knowledge graphs and study to design a new model architecture, specialized in capturing the relations among political entities.

[6] https://www.mturk.com/

[7] The six news articles are available at https://github.com/yy-ko/khan-www23.

[8] https://www.allsides.com/

Table 6: The knowledge graph (KG) completion accuracy of RotatE [62] on KG-lib and KG-con.

RotatE						
Metric	KG-lib			KG-con		
	$d = 128$	$d = 256$	$d = 512$	$d = 128$	$d = 256$	$d = 512$
MR	632.69	573.84	567.85	728.78	654.26	640.45
MRR	0.1312	0.1700	0.1859	0.1079	0.1494	0.1633
HITS@1	0.0842	0.1089	0.1209	0.0692	0.0974	0.1093
HITS@3	0.1316	0.1801	0.1985	0.1059	0.1549	0.1693
HITS@10	0.2133	0.2859	0.3148	0.1743	0.2429	0.2625

Table 7: The knowledge graph (KG) completion accuracy of ModE [71] on KG-lib and KG-con.

ModE						
Metric	KG-lib			KG-con		
	$d = 128$	$d = 256$	$d = 512$	$d = 128$	$d = 256$	$d = 512$
MR	690.14	622.40	645.23	777.11	740.88	723.90
MRR	0.1312	0.1700	0.1859	0.1128	0.1354	0.1501
HITS@1	0.0842	0.1089	0.1209	0.0685	0.0801	0.0913
HITS@3	0.1316	0.1801	0.1985	0.1127	0.1404	0.1567
HITS@10	0.2133	0.2859	0.3148	0.1981	0.2458	0.2648

Table 8: The knowledge graph (KG) completion accuracy of HAKE [71] on KG-lib and KG-con.

HAKE						
Metric	KG-lib			KG-con		
	$d = 128$	$d = 256$	$d = 512$	$d = 128$	$d = 256$	$d = 512$
MR	593.76	597.58	606.03	694.30	684.55	685.92
MRR	0.1474	0.1688	0.1787	0.1311	0.1498	0.1639
HITS@1	0.0904	0.1102	0.1167	0.0831	0.0992	0.1120
HITS@3	0.1541	0.1761	0.1895	0.1348	0.1550	0.1704
HITS@10	0.2595	0.2844	0.3013	0.2205	0.2434	0.2612

A.3 Reliability of Experiments

In this section, we verify the reliability of the experimental results, used in our empirical evaluation (EQ1. Accuracy in Section 4.2).

Evaluation protocol. As mentioned in Section 4.2, we compare the model accuracy of KHAN with the experimental results of the seven baseline methods, which have been reported in [69], on the two widely used datasets (i.e., SemEval and AllSides-S). To evaluate the reliability of the accuracy of KHAN, we (1) implement the five baseline language models, using their available codes, with a softmax layer for the final political stance prediction, (2) perform the baseline methods on the SemEval and AllSides-S datasets, (3) measure their political stance prediction accuracies, and (4) compare the results (i.e., Validation Acc.) with the reported results (i.e., Reported Acc.). For KGAP [13] and KCD [69], however, we cannot obtain the results because some parts of their source codes are not provided. More specifically, the source code to generate the cue embeddings for KCD and the source code to generate the knowledge embeddings for KGAP are missing, respectively.[9].

Result and analysis. Tables 9 and 10 show the averaged accuracy and standard deviation of each method and the reported results [69] on SemEval and AllSides-S. The results that we obtain (i.e., Validation Acc.) are quite similar as (sometimes higher than) those reported in [69] (i.e., Reported Acc.). Based on these results, we believe that our evaluation protocol and experimental results could be justified and reliable. As a result, considering that KHAN consistently outperforms all baseline methods with a very low standard deviation, our experimental results demonstrate the superiority of KHAN over existing political stance prediction methods.

Table 9: Comparison of our experimental results with the reported results [69] on SemEval (The bold font indicates the results better than the reported results).

Method	SemEval	
	Validation Acc.	Reported Acc.
Word2Vec	**0.7076** ± 0.0104	0.7027
GloVe	**0.8077** ± 0.0251	0.8071
ELMo	0.8666 ± 0.0197	0.8678
BERT	**0.8769** ± 0.0156	0.8692
RoBERTa	**0.8923** ± 0.0112	0.8708
KGAP	N/A	0.8956
KCD	N/A	0.9087
KHAN-RotatE	0.9426 ± 0.0258	N/A
KHAN-HAKE	0.9395 ± 0.0290	N/A
KHAN-ModE	0.9521 ± 0.0183	N/A

Table 10: Comparison of our experimental results with the reported results [69] on AllSides-S (The bold font indicates the results better than the reported results).

Method	AllSides-S	
	Validation Acc.	Reported Acc.
Word2Vec	**0.4977** ± 0.0082	0.4858
GloVe	0.6978 ± 0.0204	0.7101
ELMo	0.8085 ± 0.0178	0.8197
BERT	0.8201 ± 0.0101	0.8246
RoBERTa	**0.8682** ± 0.0081	0.8535
KGAP	N/A	0.8602
KCD	N/A	0.8738
KHAN-RotatE	0.9151 ± 0.0105	N/A
KHAN-HAKE	0.9216 ± 0.0041	N/A
KHAN-ModE	0.9256 ± 0.0098	N/A

[9] KGAP: https://github.com/BunsenFeng/news_stance_detection, KCD: https://github.com/Wenqian-Zhang/KCD