KW-ATTN: Knowledge Infused Attention for Accurate and Interpretable Text Classification

Hyeju Jang[1, 2], Seojin Bang[3], Wen Xiao[1], Giuseppe Carenini[1], Raymond Ng[1], Young Ji Lee[4]
[1]Department of Computer Science, University of British Columbia
[2]British Columbia Centre for Disease Control
[3]School of Computer Science, Carnegie Mellon University
[4]School of Nursing, University of Pittsburgh
{hyejuj, xiaowen3, carenini, rng}@cs.ubc.ca
seojinb@cs.cmu.edu, leeyoung@pitt.edu

Abstract

Text classification has wide-ranging applications in various domains. While neural network approaches have drastically advanced performance in text classification, they tend to be powered by a large amount of training data, and interpretability is often an issue. As a step towards better accuracy and interpretability especially on small data, in this paper we present a new knowledge-infused attention mechanism, called KW-ATTN (KnoWledge-infused ATTentioN) to incorporate high-level concepts from external knowledge bases into Neural Network models. We show that KW-ATTN outperforms baseline models using only words as well as other approaches using concepts by classification accuracy, which indicates that high-level concepts help model prediction. Furthermore, crowdsourced human evaluation suggests that additional concept information helps interpretability of the model.

1 Introduction

Text classification is a fundamental Natural Language Processing (NLP) task which has wide-ranging applications such as topic classification (Lee et al., 2011), fake news detection (Shu et al., 2017), and medical text classification (Botsis et al., 2011). The current state-of-the-art approaches for text classification use Neural Network (NN) models. When these techniques are applied to real data in various domains, there are two problems. First, neural approaches tend to require large training data, but it is often the case that large training data or pretrained embeddings are not available in domain-specific applications. Second, when text classification is applied in real life, not only the accuracy, but also the interpretability or explainability of the model is important.

As a way to improve interpretability as well as accuracy, incorporating high-level concept information can be useful. High-level concepts could help interpretation of model results because concepts summarize individual words. The concept "*medication*" would be not only easier to interpret than the words "*ibuprofen*" or "*topiramate*" but also contributes to understanding the words better. In addition, higher-level concepts can make raw words with low frequency more predictive. For instance, the words "*hockey*" and "*archery*" might not occur in a corpus frequently enough to be considered important by a model, but knowing that they belong to the concept "*athletics*" could give more predictive power to the less frequent individual words depending on the task, because the frequency of the concept "*athletics*" would be higher than individual words.

In this paper we present a new approach that incorporates high-level concept information from external knowledge sources into NN models. We devise a novel attention mechanism, KW-ATTN, that allows the network to separately and flexibly attend to the words and/or concepts occurring in a text, so that attended concepts can offer information for predictions in addition to the information a model learns from texts or a pretrained model. We test KW-ATTN on two different tasks: patient need detection in the healthcare domain and topic classification in general domains. Data is annotated with high level concepts from external knowledge bases: BabelNet (Navigli and Ponzetto, 2012) and UMLS (Unified Medical Language System) (Lindberg, 1990). We also conduct experiments and analyses to evaluate how high-level concept information helps with interpretability of resultant classifications as well as accuracy. Our results indicate that KW-ATTN improves both classification accuracy and interpretability.

Our contribution is threefold: (1) We propose a novel attention mechanism that exploits high-level concept information from external knowledge bases, designed for providing an additional layer of interpretation using attention. This attention

Proceedings of Deep Learning Inside Out (DeeLIO):
The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 96–107

mechanism can be plugged in different architectures and applied in any domain for which we have a knowledge resource and a corresponding tagger. (2) Experiments show KW-ATTN makes statistically significant gains over a widely used attention mechanism plugged in RNN models and other approaches using concepts. We also show that the attention mechanism can help prediction accuracy when added on top of the pretrained BERT model. Additionally, our attention analysis on patient need data annotated with BabelNet and UMLS indicates that choice of external knowledge impacts the model's performance. (3) Lastly, our human evaluation using crowdsourcing suggests our model improves interpretability.

Section 2 relates prior work to ours. Section 3 explains our method. Section 4 evaluates our model on two different tasks in terms of classification accuracy. Section 5 describes our human evaluation on interpretability. Section 6 concludes.

2 Related Work

2.1 Knowledge-infused Neural Networks

There has been a growing interest in incorporation of external semantic knowledge into neural models for text classification. Wang et al. (2017) proposed a framework based on convolutional neural networks that combines explicit and implicit representations of short text for classification by conceptualizing a short text as a set of relevant concepts using a large taxonomy knowledge base. Yang and Mitchell (2017) proposed KBLSTM, a RNN model that uses continuous representations of knowledge bases for machine reading. Xu et al. (2017) incorporated background knowledge with the format of entity-attribute for conversation modeling. Stanovsky et al. (2017) overrided word embeddings with DBpedia concept embeddings, and used RNNs for recognizing mentions of adverse drug reaction in social media.

More advanced neural architectures such as transformers has been also benefited by external knowledge. (Zhong et al., 2019) proposed a Knowledge Enriched Transformer (KET), where contextual utterances are interpreted using hierarchical self-attention and external commonsense knowledge is dynamically leveraged using a context-aware affective graph attention mechanism. ERNIE (Zhang et al., 2019) integrated entity embeddings pretrained on a knowledge graph with corresponding entity mentions in the text to augment the text representation. KnowBERT (Peters et al., 2019) trained BERT for entity linkers and language modeling in a multitask setting to incorporate entity representation. K-BERT (Liu et al., 2020) injected triples from knowledge graphs into a sentence to obtain an extended tree-form input for BERT.

Although all these prior models incorporated external knowledge into advanced neural architectures to improve model performance, they didn't pay much attention to interpretability benefits. There have been a few knowledge-infused models that considered interpretability. Kumar et al. (2018) proposed a two-level attention network for sentiment analysis using knowledge graph embedding generated using WordNet (Fellbaum, 2012) and top-k similar words. Although this work mentions interpretability, it did not show whether/how the model can help interpretability. Margatina et al. (2019) incorporated existing psycho-linguistic and affective knowledge from human experts for sentiment related tasks. This work only showed attention heatmap for an example.

Our work is distinguished from others in that KW-ATTN is designed in consideration of not only accuracy but also interpretability of the model. For this reason, KW-ATTN allows separately and flexibly attending to the words and/or concepts so that important concepts for prediction can be included in prediction explanations, adding an extra layer of interpretation. We also perform human evaluation to see the effect of incorporating high-level concepts on interpretation rather than just showing a few visualization examples.

2.2 Interpretability

Interpretability is the ability to explain or present a model in an understandable way to humans (Doshi-Velez and Kim, 2017). This interpretability is beneficial for developers to understand the model, help identify and possibly fix issues with the model, or to enhance the model. It is crucial for application end users because knowing explanations or justifications behind a model's prediction can further assist in decision making or the task at hand.

To provide interpretability, researchers have used inherently interpretable models such as sparse linear regression models, decision trees, or rule sets. These models are generally useful for simple prediction tasks, yet it is difficult to apply them to complicated tasks. To interpret complex models used for complex tasks, one can examine how prediction

changes between two different inputs (Shrikumar et al., 2017; Lundberg and Lee, 2017) or by locally perturbing an input (Ribeiro et al., 2016). However, a recent and popular method in NLP has been the use of an attention mechanism, which was found to be effective in helping interpret complex models by highlighting which inputs are informative to prediction (Wang et al., 2016; Lin et al., 2017; Ghaeini et al., 2018; Seo et al., 2016).

Along the lines of work using attention for interpretation, our model improves attention-based interpretability by using high-level concept information. To our knowledge, no prior work used external high-level concept information for better interpretability.

3 Our Approach

3.1 External Knowledge Bases

We automatically annotate data with high-level concepts from two knowledge bases: BabelNet and UMLS.

3.1.1 BabelNet

BabelNet (Navigli and Ponzetto, 2012) is a constantly growing semantic network which connects concepts and named entities in a large network of semantic relations, currently made up of about 16 million entries, called Babel synsets. In our study, we use the hypernyms of Babel synsets as additional higher-level concept information for the raw words or phrases in text. We first map texts with concepts in Babel synsets using an entity linking toolkit, Babelfy (Moro et al., 2014), and then retrieve hypernyms, high-level concepts, of the concepts using BabelNet APIs. Table 1 shows example annotations for the sentence "My mom was diagnosed with stage 3 ovarian cancer."

Expression	BabelNet Concepts
"Mom"	*mother*
"diagnosed"	*analyze*
"state"	*state*
"ovarian cancer"	*disease*

Table 1: Babelfy annotations for BabelNet concepts

3.1.2 Unified Medical Language System (UMLS)

We also exploit an external medical ontology, the UMLS (Lindberg, 1990), for a comparison with BabelNet for the patient need task. The UMLS is a high-level ontology for organizing a great number of concepts in the biomedical domain, which provides unified access to many different biomedical resources. On top of the UMLS, the UMLS semantic network (McCray, 2003) implements an upper-level conceptual layer for all UMLS concepts. This semantic network categorizes all concepts in the UMLS into 134 semantic types and provides 54 links between the semantic types to represent relationships in the biomedical domain.

We use the semantic types of the UMLS semantic network as additional higher-level concepts because it can abstract more fine clinical concepts that exist across much larger medical ontologies such as UMLS, SNOMED (Benson, 2010), and ICT-10(Organization et al., 2017). To obtain the semantic types, we annotate raw text by using MetaMap. Table 2 shows an example from MetaMap. Note that the automatic annotation can be noisy (e.g., incorrect semantic types for "mom" in the example).

Expression	UMLS Semantic Type
"Mom"	*Quantitative Concept*
"Diagnosed"	*Diagnostic Procedure*
"Stage 3 ovarian cancer"	*Neoplastic Process*

Table 2: MetaMap annotations for UMLS concepts

3.2 Incorporating High-Level Concepts

To incorporate high-level concept information into a NN model, we design a new attention mechanism, KW-ATTN, which allows giving separate but complementary attentions to a word and its corresponding concept. To test KW-ATTN, we choose a one-level RNN architecture with an attention mechanism (1L), a hierarchical RNN architecture with an attention mechanism (2L) as in Hierarchical Attention Network (HAN) (Yang et al., 2016), and a pretrained BERT (Devlin et al., 2018). Our 2L model architecture is shown in Figure 1. The whole architecture begins with words in each sentence as input. They are embedded and encoded using a word encoder, and then the resulting hidden representations move forward to a word-concept attention layer after being concatenated with the corresponding concept embeddings. This part is different from common RNN architectures for text classification, where only the hidden representations from the word encoder are used for a word-level attention layer. Then, the output of this attention layer is used in the next phase, a sentence encoder in case of 2L, and a final layer in case of

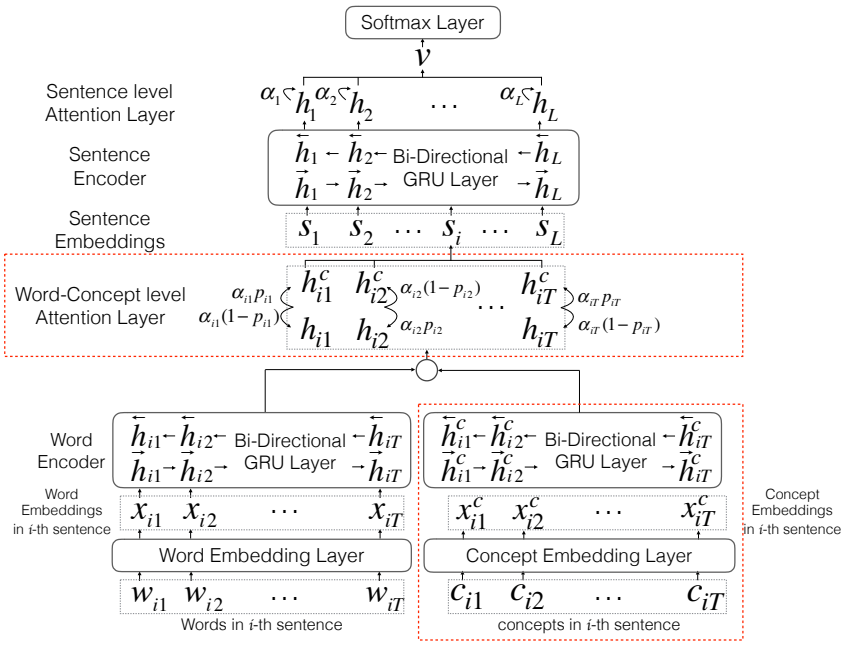

Figure 1: Overview of KW-ATTN (in red) when plugged in HAN (2L). KW-ATTN 1L does not have the sentence embeddings, sentence encoder, and sentence level attention layers. KW-BERT replaces the word encoder with a pretrained BERT model.

1L. When KW-ATTN is applied to BERT (KW-BERT), the word encoder using RNN is replaced with BERT and then the output of KW-ATTN is feed to the final layer as in 1L.

Word and Concept Embeddings: Each word w_{it} (a one-hot vector, where $t \in \{1, \cdots, T\}$ and T_i is the number of words in the i-th sentence) is mapped to a real-valued vector x_{it} through an embedding matrix W_e by $x_{it} = W_e w_{it}$. To use high-level concepts, each concept c_{it} (a one-hot vector) corresponding to word w_{it} is also mapped to x^c_{it} through an embedding matrix W_{ec} by $x^c_{it} = W_{ec} c_{it}$. When a word is not mapped into a concept, we map the concept vector to a *no-concept* vector.

Word and Concept Encoders: We encode T words in each sentence i using a word encoder. The corresponding T concepts are also encoded using a concept encoder. We use a bi-directional GRU (Cho et al., 2014) to build a representation for the t-th word and concept in the sentence i, denoted as h_{it} and h^c_{it} as follows:

$$\overrightarrow{h}_{it} = \overrightarrow{GRU}(x_{it}), \quad \overleftarrow{h}_{it} = \overleftarrow{GRU}(x_{it}),$$
$$h_{it} = [\overrightarrow{h}_{it}, \overleftarrow{h}_{it}],$$
$$\overrightarrow{h}^c_{it} = \overrightarrow{GRU}(x^c_{it}), \quad \overleftarrow{h}^c_{it} = \overleftarrow{GRU}(x^c_{it}),$$
$$h^c_{it} = [\overrightarrow{h}^c_{it}, \overleftarrow{h}^c_{it}].$$

where $t \in \{1, \cdots, T\}$, and T_i is the number of words in the i-th sentence. Note that we obtain a representation that summarizes the information of the whole sentence around the t-th word w_{it} by concatenating the forward hidden state $\overrightarrow{h}_{it}$ and the backward hidden state $\overleftarrow{h}_{it}$.

Word-Concept Attention: In this stage, the output from the word encoder h_{it} and the corresponding concept output h^c_{it} are combined by going through a word-concept level attention layer. This layer consists of two attention levels. One is an attention vector α_{it} that tracks the importance of a combined word-concept, which we call "combined" attention. The other attention vector we call "balancing" attention p_{it} is for flexibly incorporating concept information into the model. The balancing attention is introduced to give attention complementarily to both word and concept because the importance of a word or concept can differ at times. For example, when "football" is attended, we don't know if "football" itself is important for the prediction, or "football", "tennis", and all others together are important. Additionally, this balancing attention helps the model to be more robust to noisy concepts that may be caused by automatic annotation.

In detail, each position in a sentence includes a word and its corresponding concept. For each position, combined attention α is assigned, which

represents attention to the position (both word and concept). Within each position, balancing attention p is assigned to a concept and its complement $1-p$ is assigned to the corresponding word. As seen in Figure 1, α_{it} represents the contribution of the position t (both the t-th word and its concept) to the meaning of the sentence i in the sentence, while $1-p_{it}$ represents a weight on the word and p_{it} represents a weight on the word's concept. Hence, $\alpha_{it}(1-p_{it})$ and $\alpha_{it}p_{it}$ represent the contribution of the t-th word and concept to the sentence i, respectively. This attention mechanism using combined and balancing attentions enables us to give separate but complementary attentions to the word and concept. In addition, we set p_{it} as 0 when a word does not have a corresponding concept because in this case the model should attend only the word. The new attention mechanism is as follows:

$$
\begin{aligned}
u_{it} &= \tanh(W_\alpha[h_{it}, h^c_{it}] + b_\alpha) \\
p_{it} &= \mathrm{sigmoid}(w_p[h_{it}, h^c_{it}] + b_p) \\
\alpha_{it} &= \frac{\exp\left(u_{it}^T u_\alpha\right)}{\sum_t \exp\left(u_{it}^T u_\alpha\right)} \\
s_i &= \sum_t \alpha_{it}\left((1-p_{it})h_{it} + p_{it}h^c_{it}\right)
\end{aligned}
$$

where W_α, b_α, w_p, b_p and u_α are the model parameters. s_i is a representation for the i-th sentence.

s_i is used as an input to the next layer, the sentence encoder in case of 2L (HAN). Then, the sentence representations h_i go through the sentence level attention layer, and build a document vector v, as shown in Figure 1. In case of a 1L model or a BERT model, all the words in the document are treated as one single sentence. Then, there is a single representation s_1, which is equivalent to the document vector v in the 2L case.

Finally, based on this vector v, classification probability for each class is computed in the final layer.

4 Experiments

KW-ATTN is evaluated on two different datasets for patient need detection (need dataset) (Jang et al., 2019) and topic classification (Yahoo answers) (Zhang et al., 2015). We use different tasks to more broadly demonstrate the benefits of our approach.

4.1 Data

Patient need detection: This dataset is for detecting patient need in posts from an online cancer discussion forum. We use the health information need data for binary classification (450 positive samples out of 853). Although this dataset is quite small, we choose to use it because RNN approaches showed effectiveness (Jang et al., 2019) and it is a dataset we can compare the effect of general knowledge graph and domain-specific medical ontology. We build two different concept annotations with BabelNet and UMLS.

Yahoo answers: This dataset is for topic classification. It includles 10 different topics such as Society & Culture and Sports. To generate a dataset that is still small but one order of magnitude bigger than the need dataset, we randomly select 10,000 instances of the dataset enforcing a balanced dataset (1,000 instances per topic), and annotate them with BabelNet concepts.

The data statistics of our concept annotated datasets are summarized in Table 3. The ratios of words that match concepts are 6.6%(the need dataset with BabelNet), 36.3%(the need dataset with UMLS), and 8.9%(Yahoo answers). In all our experiments, we perform 10-fold cross-validation ten times. For each run, we use 80% of data for training, 10% for development, and 10% for test.

4.2 Experiment Settings

We compare our KW-ATTN 1L and 2L with a widely used attention mechanism leveraging only words (Yang et al., 2016; Ying et al., 2018). We call it **ATTN**. In addition, we use other proven approaches that leverage concept information: **Concept-replace** uses input documents where raw words are replaced with the corresponding BabelNet/UMLS high-level concepts when the mappings are available, as in (Stanovsky et al., 2017; Magumba et al., 2018). **Concept-concat** uses concatenation to combine word and concept embeddings, as in (Wang et al., 2017; Zhou et al., 2018). **Attn-concat** uses concatenation to combine a concept embedding and a hidden representation of word and use ATTN. **Attn-gating** uses a gate mechanism to select salient features of a hidden word representation, conditioned on the concept information. Both Attn-concat and Attn-gating are state-of-the-art presented by Margatina et al. (2019). All these methods are tested in 1L and 2L settings.

The parameters for RNN models are tuned on

Data	Classes	#D	#S	#W	#C(D)	#C(S)	Voca(W)	Voca(C)
Need-BN	2	853	19.2	11.0	13.9	0.7	12,484	629
Need-UMLS	2	853	19.2	11.0	75.6	6.15	12,484	118
Yahoo answers	10	10,000	7.9	12.9	10.1	1.3	65,003	3,816

Table 3: Data summary statistics. Need-BN: need dataset with BabelNet concepts, Need-UMLS: need dataset with UMLS concepts, #D: # of documents, #S: average # of sentences per document, #W: # of words per sentence, #C(D): # of annotated concepts per document, #C(S): # of annotated concepts per sentence, Voca(W): word vocabulary size, Voca(C): concept vocabulary size.

	Yahoo answers		Need BN		Need UMLS	
Model	1L	2L	1L	2L	1L	2L
ATTN	.557	.574	.706	.684	.706	.684
Concept-replace	.560	.563	.698	.671	.699	.676
Concept-concat	.569	.571	.664	.602	.702	.661
Attn-concat (Margatina et al., 2019)	.585	.577	.669	.669	.709	.681
Attn-gating (Margatina et al., 2019)	.593	.577	.712	.587	.679	.631
KW-ATTN	**.605***	**.597***	**.721***	**.692***	**.727***	**.703***

Table 4: Comparison of KW-ATTN against baselines for 1-level (1L) and 2-level (2L) networks, in terms of F1 macro scores. *: indicates statistically significant improvement over the next best model via t-test ($p < 0.05$).

development data in the following ranges: word embedding dimension: 25, 50, 100, 200, GRU size: 10, 25, 50, learning rate: 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, and 0.0001. The word embeddings are initialized randomly, and concept embeddings are initialized using pretrained concept embeddings trained on English web data and BabelNet semantic network, SW2V (Mancini et al., 2016).[1] We randomly initialize word embeddings rather than using pretrained embeddings because our model often uses phrases recognized by knowledge resources, and they are usually not part of pretrained embeddings. We optimize parameters using Adam (Kingma and Ba, 2014) with epsilon 1e-08, decay 0.0, a mini-batch size of 32, and the loss function of negative log-likelihood loss. We use early-stopping.

In addition, we also conduct experiments with pre-trained BERT Word Encoder (**KW-BERT**) to see if injecting concept also helps the model trained on large-scale corpora. We use the 'bert-base-uncased' model, and the dimension of Concept bi-GRU is 384, making the concept representation the same dimension of BERT word representations. We show both the results from frozen models and fine-tuned models.The frozen models do not update parameters of pretrained models, i.e., they use pre-trained contextualized embeddings without fine-tuning. In contrast, fine-tuned BERT or KW-BERT are adapted to the target task. The learning rates for learning frozen models and fined-tuned models are 2e-3 and 1e-6, respectively.

4.3 Experiment Results

The results are shown in Table 4. First, we observe that 2L models do not perform better than 1L models. This could be because 2L models are too large for the data sizes, especially for the need data. It could indicate that the document itself is not too long to put in one RNN, and the sentence boundary might not be necessary for the classification. Second, using concept information alone does not perform well in general, which indicates that concept information alone is not sufficient. Using word and concept information together (concept-concat) also do not always result in a gain of performance. Third, Attn- models generally perform better than simpler Concept- models. However, KW-ATTN significantly improves over all other models for both tasks, indicating the effeteness of our mechanism.

In addition, Table 4 shows that for the need task, while both types of concepts help the prediction, UMLS concepts help slightly more. This suggests that choosing the right knowledge resource, especially for domain specific tasks, is critical for prediction performance.

To see the effect of data size on the model, we

[1] We also tried SW2V Wiki, SensEmbed (Iacobacci et al., 2015) and SENSEMBERT (Scarlini et al., 2020) pretrained embeddings, but SW2V WEB slightly outperformed others (no statistical significance).

compare KW-ATTN and ATTN across different data sizes of Yahoo reviews (Table 5). KW-ATTN models significantly outperform ATTN models consistently. However, as the data size becomes larger, performance gains, while still significant, diminish, showing that, in this domain, our method is more effective when the data is smaller.

	1L		2L	
Data size	ATTN	KW-ATTN	ATTN	KW-ATTN
2,500	.460	**.523*** (+.063)	.479	**.516*** (+.037)
5,000	.527	**.561*** (+.034)	.539	**.555*** (+.016)
10,000	.557	**.605*** (+.048)	.574	**.585*** (+.011)
20,000	.611	**.634*** (+.023)	.618	**.621*** (+.003)
30,000	.624	**.645*** (+.021)	.631	**.635*** (+.004)

Table 5: F1 macro scores by data size in Yahoo answers. * indicates statistically significant improvement over corresponding ATTN model via t-test ($p < 0.05$).

Table 6 shows the comparison between BERT and KW-BERT. We can see that additional concept information substantially improves the performances on both datasets in case of frozen models whereas it only improves the performance on the need dataset when fine-tuned. The results from the frozen models indicate that the encoded concepts provide complementary information to BERT. However, when fine-tuned, KW-BERT outperforms BERT only on the Need dataset. This could be because a BERT model itself is learnt on Wikipedia, which may lack knowledge on medicine. Although BERT learns task-specific knowledge during fine-tuning, but the data is small and additional high-level concept information still helps. This may suggest that KW-BERT could be more beneficial for small data problems in domains that require more expert knowledge than Wikipedia can provide.

We can also notice that the frozen models poorly perform on the Need dataset compared with RNN models (Table 4) whereas they drastically outperform on the Yahoo dataset. This could be because the documents in the Need dataset are conversational coming from an online forum, which are markedly different from the Wikipedia dataset on which BERT is trained. We can see that when fine-tuned, both BERT and KW-BERT beat RNN models, which suggests that finetuning allows learning task/domain specific information.

Attention Analysis: To better understand why UMLS concepts help more on the need dataset, we draw the distributions of concept attentions in models with both annotations in Figure 2. Interestingly, for the average attention of each concept,

	Yahoo answers		Need UMLS	
Model	Frozen	Finetuned	Frozen	Finetuned
BERT	.652	**.698**	.585	.735
KW-BERT	**.701***	.695	**.652***	**.744***

Table 6: Comparison of BERT baseline and BERT with KW-ATTN (KW-BERT), in terms of F1 macro scores. *: indicates statistically significant improvement over the corresponding BERT baseline via t-test ($p < 0.05$).

the attention for the model using BabelNet annotations is greater than the model using UMLS annotations. However, the max attention of each concept is greater for UMLS annotations than for BabelNet annotations, which indicates that UMLS concepts are more actively used. Additionally, attentions from the model using UMLS concepts show lower variance. This result indicates that the model using UMLS concepts assigns a similar attention to each concept whereas the model using BabelNet concepts sometimes assigns small or large attentions to concept. In other words, the model using UMLS concepts consistently select a concept to attend whereas the model using BabelNet concepts is less consistent. Intuitively, this makes sense as the UMLS concepts are domain specific to the task of health information need detection.

5 Human Evaluation on Interpretability

We use human evaluation to see whether additional high-level concept information given by KW-ATTN can be beneficial for interpretation. We compare top-ranked attended words/concepts by KW-ATTN with top-ranked attended words by ATTN. We use Amazon Mechanical Turk (MTurk). Since we use crowdsourcing, we conduct evaluation only on the Yahoo reviews dataset for topic classification, which covers general domains.

5.1 Experiment Design

For each Human Intelligence Task (HIT) in MTurk, we provide a prediction and its explanation for a text, generated from either KW-ATTN 1L or ATTN 1L.[2] We use 1L because one attention layer is simpler to interpret. Then, we ask whether MTurkers would assign the given topic to the text based on the given explanation. Only one explanation is randomly given, and which model the explanations is from is not shown to MTurkers. Additionally, we ask them to rate their confidence in their answer.

[2]The screenshot of the MTurk user interface can be found in the Appendix.

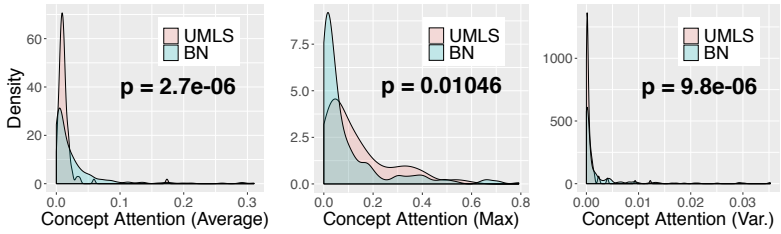

Figure 2: Distributions of concept attentions for the two annotations for patient need detection: UMLS and BabelNet (BN). For each concept, average (left), maximum (middle), variance (right) of attention values from all occurrences are used.

Explanation Type	Example
No concept	"java, yields, best, language, results, built"
KW same number	"java as a(n) object-oriented_programming_language, ide as a(n) application, php as a(n) free_software, swing, best, looking"
KW same length	"java as a(n) object-oriented_programming_language, ide as a(n) application, php as a(n) free_software"
KW replacement	"object-oriented_programming_language, application, free_software, swing, best, looking"

Table 7: Examples of different types of explanations used for human evaluation.

We assume that attention can be used for prediction explanations based on (Wiegreffe and Pinter, 2019; Serrano and Smith, 2019). We choose to ask about the validity of a given prediction unlike prior work that asked to guess a model's prediction based on an explanation (Nguyen, 2018; Chen et al., 2020). Although we acknowledge that the model's prediction may bias the annotators, we choose this approach since humans have high-level concepts as background knowledge. Humans do not require external additional concept information for guessing a correct topic label among multiple topic options especially when the given topic options are distinct from each other. For example, although the high-level concept "athletics" is not given for the word "baseball" in an explanation, humans would not have a problem with classifying it into the *sports* category when given topic options are *sports* and *music*. However, high-level concepts may help users to have more confidence when interpreting the explanation for a given topic. Therefore, we evaluate users' trusts about the system indirectly by requesting them to assess a given topic based on an explanation and rate their confidence.

The top 6 ranked features (words and concepts) with the highest attention weights are selected as an explanation. The high-level concept of a word is included in the explanation as the format of "[word] as a(n) [concept]" only when the balancing weight, p, for the concept is non-zero (See Section 3.2). We remove stopwords and punctuations from explanations.

Four different types of explanations are given to MTurkers and compared in our analysis as shown in Table 7. A **no-concept** explanation consists of 6 words. A **KW-same-number** explanation also contains 6 words and their corresponding concepts if they exist. A **KW-same-length** is composed of 3 words and their corresponding concepts if they exist. A **KW-replacement** consists of 6 words or concept. When a word has a lower attention value than its corresponding concept according to the p attention value, it is replaced by its concept in the explanation. Note that **KW-** explanations are all from the same model using KW-ATTN, and **no-concept** explanations are from a model using ATTN.

We randomly pick 200 samples that have correct predicted labels made by both systems. To make the 200 samples, we draw 100 samples with the prediction probability higher than .90 for their predicted labels, and 100 samples with the prediction probability between .80 and .90. To balance topics, we pick equal number of samples for each topic. We do not perform the same MTurk task for incorrectly predicted samples because when a system makes an incorrect prediction, assessing interpretability is not straightfoward. There can be multiple different reasons about the wrong prediction.

For MTurk, each HIT asks questions about an explanation generated by a system for one sample, as shown in Figure 3. For each HIT, 5 MTurkers participate. We hire North American Master MTurkers with HIT acceptance rates above 98% in order to ensure high quality of the evaluation. We pay \$0.03–\$0.05 for each HIT.

5.2 Human Evaluation Results

As shown in Table 8, KW-same-number and KW-same-length explanations resulted in a significantly higher confidence in assigning given topics to explanations compared to no-concept explanations. This indicates that the additional high-level concept information from KW-ATTN is beneficial for improving interpretability. We can also observe that KW-replacement explanations improve confidence although the gain is not significant.

Explanation Type	Pred	Conf	Time
No-concept	4.70	4.15	11.31
KW-same-number	4.82	4.40*	11.64
KW-same-length	4.77	4.31*	11.37
KW-replacement	4.74	4.22	12.34

Table 8: Human evaluation results on interpretation. Pred: average # of "yes" on predicted topics, Conf: average confidence score, Time: average time taken for each HIT, *: indicates statistically significant difference over no-concept via t-test ($p < 0.05$).

It is important to note that KW-same-length and KW-replacement explanations both improve interpretability over no-concept explanations as well as KW-same-number. While KW-same-number explanations provide more information (12 at maximum in total including both words and concepts), KW-same-length and KW-replacement give the same or less amount of information compare to no-concept (6 at maximum in total). This indicates that the high-level concept information really helps.

6 Conclusion

We presented a new attention mechanism, KW-ATTN, which extends a NN model by incorporating high-level concepts. Our experiments showed that using high-level concept information improves predictive power by helping the data sparseness problem in small data. Furthermore, in our crowdsourcing experiments, we found significant improvement on the confidence of human evaluators on predictions, suggesting that our new attention mechanism provides benefits in explaining the predictions. High-level concepts provide an additional layer of information above raw words that can assist in understanding predictions. Additionally, our attention mechanism can distinguish between the importance of words vs. concepts, providing further information. We are optimistic that KW-ATTN can be applied widely.

References

Tim Benson. 2010. Snomed ct. In *Principles of Health Interoperability HL7 and SNOMED*, pages 189–215. Springer.

Taxiarchis Botsis, Michael D Nguyen, Emily Jane Woo, Marianthi Markatou, and Robert Ball. 2011. Text mining for the vaccine adverse event reporting system: medical text classification using informative feature selection. *Journal of the American Medical Informatics Association*, 18(5):631–638.

Hanjie Chen, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. *arXiv preprint arXiv:2004.02015*.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Christiane Fellbaum. 2012. Wordnet. the encyclopedia of applied linguistics.

Reza Ghaeini, Xiaoli Z Fern, and Prasad Tadepalli. 2018. Interpreting recurrent and attention-based neural models: a case study on natural language inference. *arXiv preprint arXiv:1808.03894*.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 95–105.

Hyeju Jang, Young Ji Lee, Giuseppe Carenini, Raymond Ng, Grace Campbell, and Kendall Ho. 2019. Neural prediction of patient needs in an ovarian cancer online discussion forum. In *Canadian Conference on Artificial Intelligence*, pages 492–497. Springer.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Abhishek Kumar, Daisuke Kawahara, and Sadao Kurohashi. 2018. Knowledge-enriched two-layered attention network for sentiment analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 253–258.

Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. 2011. Twitter trending topic classification. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 251–258. IEEE.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

C Lindberg. 1990. The unified medical language system (umls) of the national library of medicine. *Journal (American Medical Record Association)*, 61(5):40–42.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *AAAI*, pages 2901–2908.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.

Mark Abraham Magumba, Peter Nabende, and Ernest Mwebaze. 2018. Ontology boosted deep learning for disease name extraction from twitter messages. *Journal of Big Data*, 5(1):31.

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2016. Embedding words and senses together via joint knowledge-enhanced training. *arXiv preprint arXiv:1612.02703*.

Katerina Margatina, Christos Baziotis, and Alexandros Potamianos. 2019. Attention-based conditioning methods for external knowledge integration. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3944–3951.

Alexa T McCray. 2003. An upper-level ontology for the biomedical domain. *International Journal of Genomics*, 4(1):80–84.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1069–1078.

World Health Organization et al. 2017. International classification of diseases (icd) information sheet. 2017.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *AAAI*, pages 8758–8765.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*.

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):22–36.

Gabriel Stanovsky, Daniel Gruhl, and Pablo Mendes. 2017. Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 142–151.

Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *IJCAI*, pages 2915–2921.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into conversation modeling via recall-gate lstm. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3506–3513. IEEE.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention network. In *IJCAI International Joint Conference on Artificial Intelligence*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. *arXiv preprint arXiv:1909.10681*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

A Appendices

Figure 3 shows a screenshot of the Amazon Mechanical Turk user interface in our human evaluation.

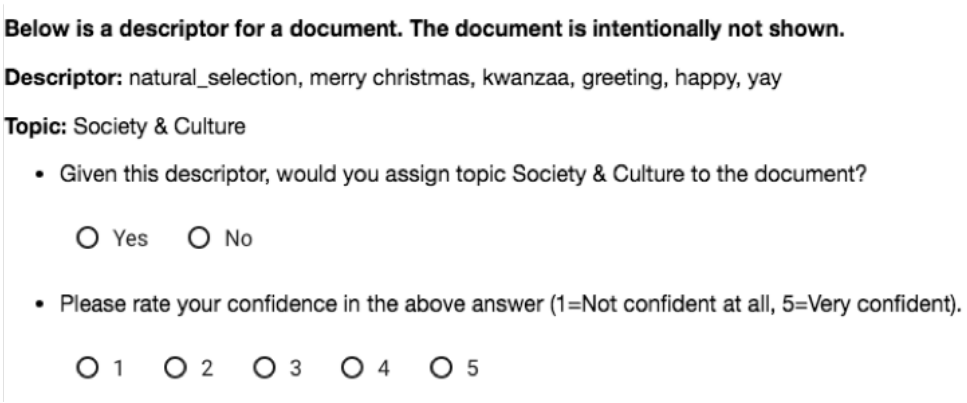

Figure 3: Our MTurk interface for human evaluation about interpretability.