Pay Attention to Categories: Syntax-Based Sentence Modeling with Metadata Projection Matrix

Won Ik Cho and **Nam Soo Kim**
Department of Electrical and Computer Engineering and INMC,
Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, Korea, 08826
wicho@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

Sentence modeling is a vital feature engineering for document classification. Various feature extraction and summarization algorithms have been adopted for efficient classification of a sentence, e.g., dense word vectors and neural network classifiers. Recently, the concept of attention for machine translation has been applied to various natural language processing (NLP) tasks and has shown significant performance. In this paper, we take a look at the syntactic categories of the words, to make up a metadata projection matrix that assigns strong restrictions on determining the attention weight. Unlike conventional attention models, which are considered as a division of location-based approaches, our model adds a selection layer to highlight categorical metadata that may appear more than once. The proposed algorithm shows improved performance compared to the baselines with the tasks in syntax-semantics, suggesting a possibility of extension to other fields such as symbolic music or bitstream analysis.

1 Introduction

Sentence modeling, which incorporates featurization and embedding, has been widely studied from short utterances to large-scale documents. Its usefulness and broad applicability have been proven with various classification and regression tasks. Also, in recent years, attention models have demonstrated the significant performance of such approaches, along with deep learning techniques that have shifted the paradigm of the standard recipes.

In applying the attention models, we noted that the utility of the syntactic properties should be explored in a bit wide point of view. Like the notes in music that have corresponding chords, the observable components of a sentence are assigned syntactic categories after constituency parsing, such as noun, verb, and adjective. They are interpreted as a kind of metadata regarding each token[1], that may appear more than once in the document. We want to claim such information can be exploited in making up the attention weight, not just being adopted as input-level data. For instance, in an oxymoron identification task (Cho et al., 2017), given a sentence like "*This is a sugar-free sweet tea.*", it may be beneficial for the analysis to attend to *sugar-free* and *sweet* with a similar concentration, mainly due to their syntactic property being close to each other.

Although such syntactic properties can be represented in various ways such as tree structure and dependency, we pay attention to part-of-speech (POS), for some practicality. First of all, we already have many computationally efficient tools that can extract syntactic classes from the tokens of the sentence. Next, even though the POS tagger is not entirely accurate, the general tendency may provide sufficient information for classification. This flexibility can be supportive for the proposed model to analyze corpus with non-formal sentences such as tweets.

The proposed model differs from the usual self-attentive models in that it takes into account the information of syntactic categories while maintaining

[1] Henceforth, we interchangeably use (token-wise) categorical data, categorical metadata, and categorical information, all referring to the syntactic classes that each token belongs to.

the original form of classification that uses word vector sequence[2]. Furthermore, the model tells us how much attention we should pay to the components with specific syntactic properties, given the overall summarization of a sentence. The contribution of this study is as follows:

- We suggest a modified version of the conventional location-based attention model by inserting a simple projection layer that contains information on the syntactic categories.
- We verify the utility of the proposed scheme with widely used benchmarks and suggest further usage.

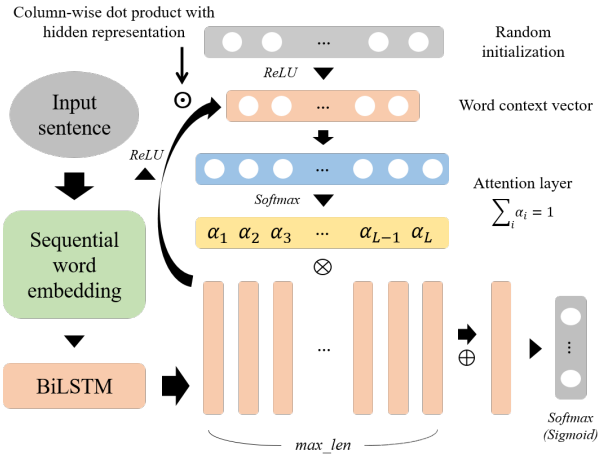

Figure 1: Descriptive diagram of attention model presented in self-attentive sentence embedding (Lin et al., 2017). The arrows in the figure indicate the flow of information. The triangles in the overall system denote the fully connectedness to the dense layer, together with the stated activation functions.

2 Related Work

2.1 Sentence embedding

Embedding a sentence into numerics is an essential process in data-driven sentence classification. Two major types of representation are widely used, namely sparse and dense.

One of the most popular sparse word representations, bag-of-words (BoW) model, is a one-hot encoding of the words in the sentence and is most commonly used for its conceptual clarity. Another well known sparse representation is the term frequency-inverse document frequency (TF-IDF), which conveys the relative importance of the terms in each document.

The main issue of BoW and TF-IDF is that they can hardly give information about the context window of each term in a sentence. Thus, count-based approaches for the local context window of words have been studied, as in Lebret and Collobert (2013). However, it can also be problematic because such approaches can disproportionate weight to words with large counts. They can also cause a dimensional explosion.

To cope with the above, Mikolov et al. (2013) proposed an algorithm that embeds a word into a low dimensional dense vector that involves a local context window. The real-vectorized words facilitate similarity computation between the original words and can be used to represent sentences, e.g., by averaging Le and Mikolov (2014). In Pennington et al. (2014), the advantages of the approaches in Lebret and Collobert (2013) and Mikolov et al. (2013) were combined.

2.2 Modeling techniques in classification

In sentence classification, basic recipes such as naive Bayes, decision trees, and logistic regression models were conventionally used. Among such models, the support vector machine (Cortes and Vapnik, 1995) showed quite a practical accuracy.

However, ever since the computational breakthrough that had taken place in the deep neural network (DNN) system (Hinton et al., 2006), neural architectures have been adopted within the sentence classification tasks, along with the emergence of dense word vectors. Convolutional neural network (CNN), which initially came up for the image classification task (Krizhevsky et al., 2012), was successfully applied to the sentence classification task (Kim, 2014). Recurrent neural networks (RNN) (Schuster and Paliwal, 1997; Graves, 2012), which had been proposed to deal with sequential data processing, also have shown significant performance in sentence classification tasks through various forms such as gated recurrent unit (GRU) (Tang

[2]In other words, here we don't adopt attachment such as 'word/POS'.

et al., 2015) and bi-directional long short-term memory (BiLSTM) (Chen et al., 2017), comprehensively summarizing sentences into dense vectors.

Lately, attention models have been applied to the neural machine translation (Bahdanau et al., 2014) in the way of multiplying the attention vector with the decoder-encoder network matrix to generate a particular target word from the source word. It can be regarded as jointly training a weight vector augmented to a feature or hidden layers to focus on a specific part of the input feature. Driven by its conceptual clarity, it was soon applied to areas such as image captioning (Xu et al., 2015) and natural language interface (Liu et al., 2016).

In Lin et al. (2017), the self-attentive embedding (SA, Figure 1) was applied to the sentence classification, by aggregating essential attributes of the hidden layers into sentence vectors. A word context vector, which is multiplied by the higher-level representation of hidden layers in BiLSTM, is used to create attention (weight) layer with a sum equal to 1.

In detail, for $X = X_1^L$ the input token sequence, $H = H_1^L$ the hidden layers, weight W_t and bias b_t, the BiLSTM hidden layers are defined as:

$$H_t = \tanh\left(W_t\left[X_t, H_{t-1}\right] + b_t\right) \tag{1}$$

As in the right top of Figure 1, each hidden layer is multiplied with word context vector C to yield a softmax-ed attention vector α with $\sum_t \alpha_t = 1$, as:

$$\alpha_1^L = \mathrm{softmax}\left(H_1^L \odot C\right) \tag{2}$$

where $\odot$ denotes a column-wise dot product. α_1^L is further multiplied to H_1^L and is summed to be fed to the final decision layer, as a representative hidden layer output:

$$H_o = \sum_t \alpha_1^L \otimes H_1^L \tag{3}$$

where $\otimes$ denotes a column-wise multiplication of the scalar weights. In the figure, L equals to the maximum sentence length *max_len* and $\oplus$ denotes the weighted sum of the hidden layers.

Note that this basic architecture covers most of the sentence-level attention schemes that precede the contemporary self-attention models (Vaswani et al., 2017; Devlin et al., 2019). In this regard, at this point, we consider this structure suffices as a baseline to implement our scheme on, due to the assignment of attention weight being interpretable and straightforward.

3 Proposed Method

In this section, we demonstrate the concept of *Pay Attention to Categories*, or PAC structure, which can adequately reflect the categorical metadata of each token onto the attention model. It denotes an insertion of a projection matrix that incorporates the information on syntactic classes, which yields the modified attention weight that comes afterward. Materializing it accompanies three main steps, namely (a) constructing word vector sequence, (b) feature extraction for the attention source, and (c) projecting the weight that corresponds with the category of each token (or here, syntactic classes) to the attention layer.

(a) Word vector sequence can be constructed by methodologies used in general. It is briefly depicted at the bottom of Figure 2, especially step (2), where *max_len* denotes the upper limit of the sentence length regarding word count. Summarizers such as CNN and BiLSTM employ this as a feature, using sigmoid (binary case) or softmax (multi-class) as an activation function.

(b) Attention source utilizes various features extracted from the sentence. It can be TF-IDFs, averaged word vectors, or the output layer of a CNN or BiLSTM summarizer. In this paper, (bigram) TF-IDF and BiLSTM hidden layer output were adopted based on the performance. They are fed to PAC structure after passing a single dense layer with rectified linear unit (ReLU) activation, as depicted in the top of Figure 2.

(c) PAC structure consists of a layer carrying the category-wise weight (shortly a weight layer), a projection matrix, and their multiplication (the attention layer). The size of the weight layer (n_p) equals to the number of the categories that appear throughout the document.

In detail, let S be the attention source and $ReLU, hsig$ be activation functions. Then, for given n_p, we get:

$$w_p = \mathrm{hsig}(\mathrm{ReLU}(S)) \tag{4}$$

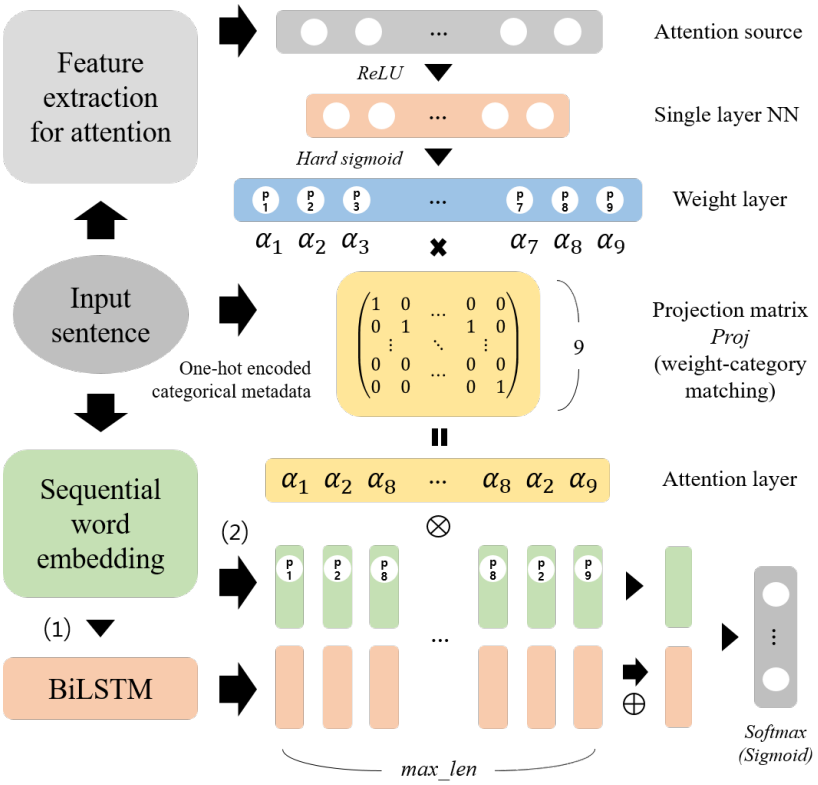

Figure 2: A Descriptive diagram for the proposed system.

On the other hand, we have a fixed projection layer which contains the syntactic information regarding each token. The matrix $Proj$ is of size (n_p, L), and each column tells the syntactic category each token belongs to. In this study, it is represented by POS. We multiply it with the former weight layer to obtain the attention layer of width L:

$$\alpha_1^L = \text{matmul}\,(w_p, \text{Proj}) \tag{5}$$

It consists of the weight corresponding to each word of the sentence and is column-wisely multiplied to either the hidden layers (*PAC-Hidden*) or the word vector sequence (*PAC-Word*). The two strategies are depicted in Figure 2, where $\times$ denotes a matrix multiplication and $\otimes$ denotes a column-wise multiplication of the attention layer to (1) the hidden layer sequence as BiLSTM output (*PAC-Hidden*), or (2) the original word vector sequence (*PAC-Word*). For *PAC-Word*, the weighted word vector sequence becomes an input of BiLSTM again.

More on the figure, to help the readers understand, we specified the number of categories (n_p = 9), as shown in the weight layer w_p. The sequence of one-hot encoded vectors of categorical metadata, $Proj$, expressed in the form of a projection matrix, conveys the weight to the attention layer, concerning the syntactic class that each column (of hidden layers or word vector sequence) incorporates. For instance, if the index regarding a word's syntactic class is 2, as in the case of the second and the second to the last, it is multiplied by the value conveyed from α_2. Note that this setting allows the repetition of the attention weight. It is worth noting that the activation function of the weight layer is set to hard sigmoid as in (4). We surmised that the hidden layer's information should be fully retained even after it is transferred to the projection matrix. Here, hard sigmoid plays a vital role, minimizing information that can be nullified in multiplication with the one-hot encoded matrix.

4 Experiment

In this section, we describe the benchmark datasets, the specific implementation scheme, and the result comparison with baselines.

4.1 Dataset

Five datasets were used in the evaluation. The specification for the datasets is displayed along with the corpus size.

Metalanguage detection (2,393) employs the corpus for English metalanguage detection (Wilson, 2012), which investigates whether a sentence contains explicit mention terms, namely with the lexicons such as 'title' or 'name'. It contains 629 *mentioned* and 1,764 *not-mentioned* instances excerpted from Wikipedia.

Irony detection (4,618) utilizes corpus recently distributed in SemEval 2018 Task 3 for ironic tweet detection (Van Hee et al., 2018). All instances (that includes emoji) in the training set and Gold test data were used. Only the binary label case was taken into account. 2,222 instances contains *irony* and 2,396 does not.

Subjectivity detection (10,000) refers to Pang and Lee (2004), which checks if the movie review contains a subjective judgment, in the view of sentiment polarity. It incorporates equally 5,000 instances for each of the *subjective* and *objective* reviews.

Stance classification (3,835) employs a part of the distributed dataset from SemEval 2016 Task 6 (Mohammad et al., 2016). The original dataset consists of the additional labels corresponding to target, stance, opinion towards and sentiment information. All instances with favor and against stances in the dataset were excerpted. Among instances with none as stance, only those not explicitly expressing opinions were taken into account. There are 1,205, 2,409, and 221 instances for *favor*, *against*, and *none* each.

Sentiment classification (20,632) utilizes the test data released in SemEval 2017 Task 4 (Rosenthal et al., 2017). It consists of 7,059 *positive*, 3,231 *negative* and 10,342 *neutral* tweets, with all instances labeled via crowd-sourcing.

4.2 Implementation

The implementation for the whole network was done with Python libraries, including NLTK (Bird et al., 2009), Scikit-learn (Pedregosa et al., 2011), and Keras (Chollet and others, 2015). In particular, POS tagging and word tokenization process employed the tools included in NLTK. Here, elaborate implementation schemes of baselines and the proposed system are presented.

4.2.1 Baselines

Features Baseline features were chosen from both sparse and dense ones. For sparse features, TF-IDFs and their bigrams were extracted. The dimension (the number of commonly used words) was fixed to 3,000 for a fair comparison, which is the same size as a multiplication of *max_len* (=30) and word vector dimension (=100). The uni/bigrams were obtained via TfidfVectorizer of Scikit-learn.

For dense features, 100-dimensional GloVe (Pennington et al., 2014) pre-trained with 27B token Twitter data[3] was adopted as a word vector dictionary, since the words thereof were expected to cover the lexicons of the task corpora. The dense static features were constructed by aggregating the vectors corresponding to every word in the sentence and normalizing it using the l_2 norm. The dense sequential features were constructed by padding the word vectors with a maximum length of 30.

Basic classifiers All evaluations were conducted using 10% test set. Non-parameter optimized linear-kernel SVM of Scikit-learn was used for sparse features (*TF-IDF-SVM*), and NN classifiers in Keras were used for dense features. NN used for the static dense features (*Averaged GloVe-NN*) consists of a single hidden layer of size hidden dim and is optimized with Adam (Kingma and Ba, 2014) of learning rate 0.0005. The network was trained with mini-batch of size 16, reducing the cross-entropy loss. The implementation toolkit, optimizer, and mini-batch size for all NN classifiers were not changed throughout the experiment. For every model, *hidden_dim* was chosen as the best case after hyperparameter tuning with 32, 64, and 128.

CNN and BiLSTM were used in the baseline sequential feature classification (*GloVe-CNN/BiLSTM*). In CNN, two single-channel convolutional layers (with 32 filters and a window of size 3) were used with a max-pooling layer in between. In BiLSTM, time-distributed hidden

[3] https://nlp.stanford.edu/projects/glove/

layers had an output size of 32*2 = 64 units.

Baseline attention model The attention adopted from Lin et al. (2017) was implemented as depicted in Figure 1. The word context vector of size *hidden_dim*, which is fully connected to a randomly initialized layer, is column-wisely dot-multiplied by the *ReLU*-activated[4] hidden representation of the sequential hidden layers of BiLSTM, also of size *hidden_dim* and length *max_len*. Note that the product layer of size *max_len* undergoes the regularization process using the softmax function (sum to 1), unlike the model proposed in this work.

The attention vector was applied to the word sequence in two different ways: by directly multiplying it to a hidden layer sequence (*SA-Hidden*), or by multiplying it to a word vector sequence (*SA-Word*). In the former case, which was suggested in the original paper, the final decision was made by investigating the weighted sum of the hidden layers. The latter case, which was supplemented to observe the tendency of each strategy, investigates the weighted word vector sequence with BiLSTM.

4.2.2 The Proposed

The proposed system extracts three input features from each sentence: *attention source, projection matrix, and word vector sequence*.

As previously mentioned, two features were adopted as attention source: TF-IDFs and BiLSTM outputs. For TF-IDFs, the sparse vector of dimension 3,000 is itself an attention source[5]. Unlike the case of TF-IDFs where the source is assigned as an input, all parameters of BiLSTM are trained jointly with the entire system.

The attention source is fully connected to the single dense layer of size hidden dim, with *ReLU* activation. Consecutively, this is fully connected to the weight layer with hard-sigmoid activation, as described in the previous section.

The size of the weight layer and the projection matrix depends on the corpus. In a corpus with n_p syntactic classes (the number of categories), a weight matrix of length n_p and a projection matrix of size (n_p , *max_len*) are obtained. Again, the emphasis is that the weight layer is optimized in the training session, but the projection layer is given as input.

Finally, the attention layer of size *max_len* appears as the product of the matrix multiplication of the weight layer and projection layer. All its entries are multiplied as the weight to each column of either the hidden layer sequence of BiLSTM (*PAC-Hidden*) or the word vector sequence (*PAC-Word*).

5 Result and Discussion

Per task characteristics The proposed system surpasses the baseline systems in tasks that are expected to be accompanied by lexical-semantic analysis, such as *META*, *IRONY*, and *SUBJ* (Table 1). Also, it was observed that the systems fit with small datasets as well, considering the significant improvement in *META* and *IRONY*. In tasks where semantics are considered much more important, such as *STANCE* and *SENT*, the proposed system showed a stable and adequate result, not an improvement in performance. This result implies that the proposed system may rather boost the performance of the tasks that utilize the existence and meaning of the lexicons thereof, than the semantic tasks that require more a latent analysis.

Source and assignment of attention We observed that the tendency regarding attention source, namely TF-IDF or BiLSTM output, is opaque and non-consistent, considering that no significant tendency is displayed. On the other hand, the contrast on *Word-* level and *Hidden-* level assignment of attention weight is quite significant per task. Especially for *META*, *IRONY*, and *SUBJ*, where the proposed methods outperform the baselines, we found that *META* highly prefers *Word*-level assignment, while *Hidden*-level assignment works better for the other two. This directly shows that *META* concerns the explicit existence of certain lexical terms, while the other two touch relatively abstract areas of lexical-semantics.

Under context-dependency Specifically, the lower performance and stochastic results in *STANCE* seem to originate in the omission of *target data* in this experiment. It is essential situational

[4] In view of performance and fair comparisons, *tanh* used in the original paper was replaced with *ReLU*.

[5] In case of *META* and *STANCE*, bigram was chosen considering the comparison result (Table 1).

F1 Score	Features	META	IRONY	SUBJ	STANCE	SENT
Sparse features	***TF-IDF***	0.5466	0.6236	0.8953	0.4316	0.5604
	Bigram TF-IDF	0.5489	0.6137	0.8944	0.4334	0.5509
Dense features	***Averaged GloVe-NN***	0.5454	0.6455	0.8845	0.3676	0.6157
	GloVe-CNN	**0.6800**	0.6613	0.9036	0.4141	0.6121
	GloVe-BiLSTM	0.6527	0.6639	0.9159	**0.4763**	0.6304
Attention	***SA-Word***	0.6363	0.6447	0.9152	0.3703	0.6297
	SA-Hidden	0.6478	**0.6771**	**0.9203**	0.4317	**0.6538**
Proposed	***TF-IDF PAC-Word***	0.7105	0.6679	0.9204	**0.4671**	0.6241
	TF-IDF PAC-Hidden	0.6535	**0.7019**	**0.9268**	0.4253	0.6329
	BiLSTM PAC-Word	**0.7261**	0.6585	0.9135	0.4332	0.6353
	BiLSTM PAC-Hidden	0.6400	0.6956	0.9259	0.4475	**0.6529**

Table 1: Performance comparison of the baselines and the proposed system. *META, IRONY, SUBJ, STANCE*, and *SENT* denote the datasets in Section 4.1, respectively. *SA-Word/Hidden* refer to the self-attentive embedding models. TF-IDF and BiLSTM coming before *PAC-Word/Hidden* represent the attention sources. The final decision of the proposed systems was also made through BiLSTM. In the baselines and the proposed models, the best scores were bolded. Underlined cases denote when the proposed system surpasses the baseline.

information in determining a stance towards someone but was not digitized in this experiment. Also, there was a shortage in the number of instances associated with none. On the other hand, for instance, in *IRONY* where situational information is essential as well, the proposed system showed an outperformance. It is assumed that in *IRONY*, hashtagged information plays a critical role (Cho et al., 2018), and accordingly, attention is given to functional parts as well.

Summary We concluded that paying attention to relatively important syntactic classes such as verbs (*META*), nouns (*IRONY*), or adjectives (*SUBJ*·*IRONY*) is advantageous in some tasks. This inference is also consistent with the consideration for polarity items (Krifka, 1995), which takes into account the relation between words of different syntactic classes. From this point of view, a suitable application of the proposed system would be a case where the categorical metadata plays a significant role in determining the labels of the data, and the pattern is relatively clear, e.g., bitstream analysis.

5.1 Visualization

The normalized attention weight of baseline and the proposed, namely *SA-Hidden* and *TF-IDF PAC-Hidden*, are visualized as Figure 3 with two excerpt sentences from *SUBJ*.

Considering the property of the dataset, it is clear that the attention should be given to the words in the sample sentences that affect the subjectivity. In the top example, the baseline model pays attention to *vile* and *tacky*, which are the subjective modifiers indicating the object *ghost ship*, while the proposed model addresses *best*, the superlative which can directly show the subjectivity of the sentence. Besides, at the bottom, the proposed model pays attention to *comedy*, which reveals the sarcastic tone, while the baseline only attends to *funniest* among the lexical candidate words.

Without a doubt, this kind of advantage in the inference partially benefits from the task being sensitive to specific sentiment items in the sentence. Nonetheless, beyond the examples above, the proposed model can stably give attention to the specific categories that seem to be important in analyzing the document. Given that this kind of consistency is sometimes threatened in the analysis of informal or non-canonical utterances, stable fixation of weight can be advantageous often. Also, we note that each category's weight varies with the content of the sentence, making the proposed model differ from hard attention.

5.2 Further Study

Beyond a simple application that considers syntactic categories as property for words, the proposed system can be extensively utilized to datasets where observable components contain metadata of a type

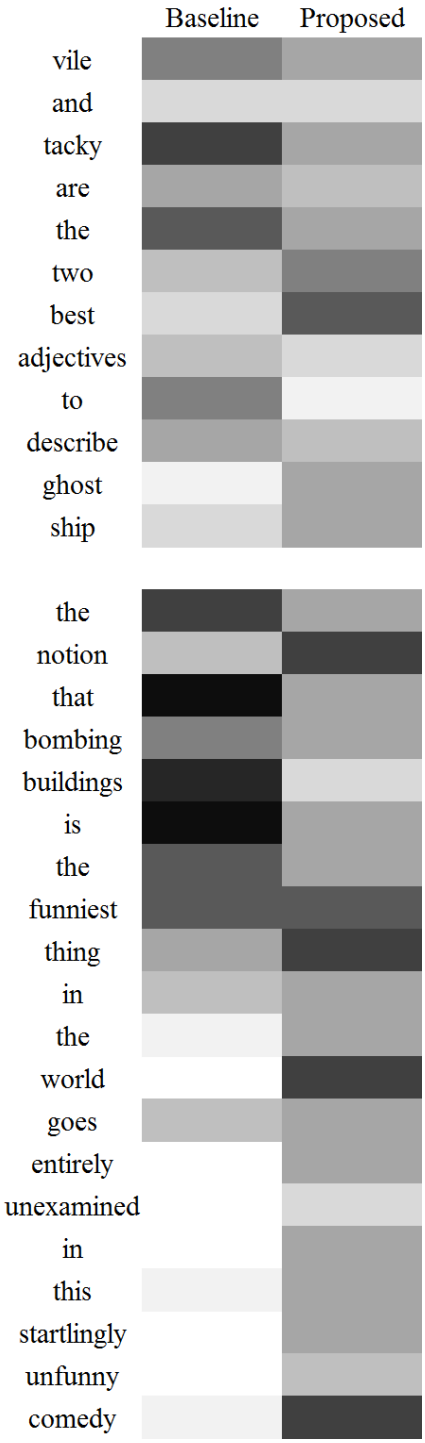

Figure 3: Visualization of the attention weight given to the subjective example sentences in *SUBJ*.

that possibly appears more than once. For example, in a paragraph or large-scale document analysis, a sentence type or document topic can be used as such information. In the field of music information retrieval, chord information can be provided to the attention model to help predict whether the type of the musical phrase (Livingstone et al., 2009) is cadence, semi-cadence, false cadence, or nothing. In acoustic event detection (Choi et al., 2017), event labels can also be used as a property to identify acoustic scenes, even in the multi-label conditions.

6 Conclusions

In this paper, the concept called *Pay Attention to Categories*, or PAC structure, was suggested for efficient sentence classification. The proposed system fully utilizes the syntactic class of each token, which is modeled in terms of POS for words, in making up a special kind of projection matrix, and employ it in building up attention weight. Its conceptual simplicity and flexibility were demonstrated with an intuitive diagram, and the validity was verified via comparison with widely used benchmarks. Beyond utilities in many NLP areas, the system is expected to have a significant role in tasks that require attention to categorical information.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00059, Deep learning multi-speaker prosody and emotion cloning technology based on a high quality end-to-end model using small amount of data).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. 2017. Improving sentiment analysis via sentence type classi-

fication using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230.

Won Ik Cho, Woo Hyun Kang, Hyun Seung Lee, and Nam Soo Kim. 2017. Detecting oxymoron in a single statement. In *Proceedings of Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 48–52.

Won Ik Cho, Woo Hyun Kang, and Nam Soo Kim. 2018. Hashcount at semeval-2018 task 3: Concatenative featurization of tweet and hashtags for irony detection. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA, June. Association for Computational Linguistics.

Inkyu Choi, Soo Hyun Bae, Sung Jun Cheon, Won Ik Cho, and Nam Soo Kim. 2017. Weakly labeled acoustic event detection using local detector and global classifier. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1735–1738. IEEE.

François Chollet et al. 2015. Keras. `https://github.com/fchollet/keras`.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

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

Alex Graves. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

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

Manfred Krifka. 1995. The semantics and pragmatics of polarity items. *Linguistic analysis*, 25(3-4):209–257.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Rémi Lebret and Ronan Collobert. 2013. Word emdeddings through hellinger PCA. *arXiv preprint arXiv:1312.5542*.

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*.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Steven R Livingstone, Emery Schubert, Janeen Loehr, and Caroline Palmer. 2009. Emotional arousal and the automatic detection of musical phrase boundaries. In *Proceedings of the International Symposium on Performance Science*, pages 445–450.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, pages 1532–1543.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in English Tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, SemEval-2018, New Orleans, LA, USA, June. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Shomir Wilson. 2012. The creation of a corpus of English metalanguage. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 638–646. Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057.