Conversational Decision-Making Model for Predicting the King's Decision in the Annals of the Joseon Dynasty

JinYeong Bak
School of Computing
KAIST
Republic of Korea
jy.bak@kaist.ac.kr

Alice Oh
School of Computing
KAIST
Republic of Korea
alice.oh@kaist.edu

Abstract

Styles of leaders when they make decisions in groups vary, and the different styles affect the performance of the group. To understand the key words and speakers associated with decisions, we initially formalize the problem as one of predicting leaders' decisions from discussion with group members. As a dataset, we introduce conversational meeting records from a historical corpus, and develop a hierarchical RNN structure with attention and pre-trained speaker embedding in the form of a, Conversational Decision Making Model (CDMM). The CDMM outperforms other baselines to predict leaders' final decisions from the data. We explain why CDMM works better than other methods by showing the key words and speakers discovered from the attentions as evidence.

1 Introduction

Decision making in groups refers to the process of making choices to resolve issues by discussing the issues with group members (Lunenburg, 2011). It has various styles based on the balance of the participation between the leader and members from autocratic, democratic, *laissez-faire* (let go) to delegation types of groups (Lewin et al., 1939; Vroom and Jago, 1988). Social psychologists note that decision making affects the group performance and the satisfaction of its members (Yang, 2010), and that leadership plays a role (Larson Jr et al., 1998). In this paper, we study the key factors that are closely related to the decision making process used by leaders.

First, we build conversational meeting records from *The Annals of the Joseon Dynasty* (henceforth referred to as the AJD), after which, we formalize our research problem as predicting leaders' decisions in conversational discussions from the data (Sec 2). The AJD consists of the records of kings who governed the Korean peninsula from

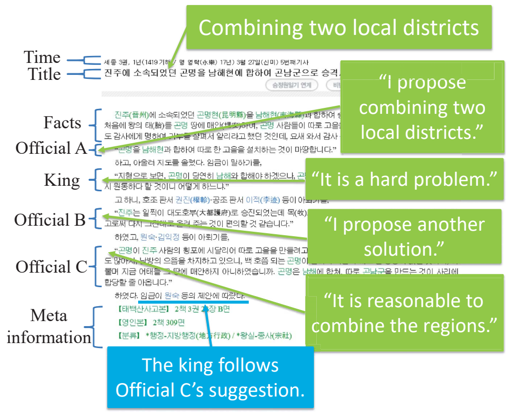


Figure 1: Screenshot and structure of an article in the annals of the Joseon dynasty

1392 to 1910. In the AJD, the kings discuss the issues with government officials and decide upon a course of action. Many discussion corpora are available such as Augmented Multi-party Interaction (AMI) (Carletta et al., 2005) which is meeting recordings as video, and are used to identify and summarize decisions in the conversation (Hsueh and Moore, 2007; Fernández et al., 2008; Bui et al., 2009). However, the AJD has more speakers than AMI, and it is a longitudinal corpus spanning over 400 years.

To predict the decisions in the corpus, we develop a model which we term the Conversational Decision-Making Model (CDMM) (Sec 3). CDMM is based on the hierarchical RNN structure with attention (Yang et al., 2016), but we add speaker information with pre-trained embedding. We also devise a way to make the speaker embedding using co-occurrence document network (Sec 3.3). In comparison with several other methods, CDMM shows the highest macro-averaged F1 score (Sec 4). We also show why CDMM works better with key words and speakers by examining the attention values (Sec 5).

Kings	Articles	Utterances	Participants
15	13,216	95,615	4,502

(a) Basic statistics of the corpus

Order	1,996	*Accept*	1,457
Approve	2,245	*Reject*	818
Disapprove	468	*Discuss*	6,214

(b) Distribution of articles over decisions

Table 1: Statistics of a conversational meeting records and king's decisions from the AJD

2 Meeting Records from the AJD

Historiographers recorded the behaviors of kings and events in the country, and compiled these records as books when the king died or abdicated the throne. Each article of the AJD consists of the time, title, body and meta-information such as categories.

Meeting articles in the AJD consist of who said what on an issue in dialogue form, and the king's decision. Figure 1 shows an example of a meeting record article[1]. In the article, the king and government officials discuss the issue of combining two local regions. The king asks for a solution to the issue from the officials, and they state their opinions. At the end of the article, the king decided to follow official C's suggestion to solve this issue.

We build a corpus from the AJD using the following process. We crawl the AJD website to retrieve the documents and select articles that have three or more speakers per document. We identify the king's final decision in each article by examining the final sentence and the title as summarized by historians. We initially determine whether or not the final sentence of the subject is that by a king, as some issues are dealt with by others, such as the king's mother. We also extract the verbs in the final sentence and the title that indicates the decisions. From these, we categorize each king's decisions into six types: *Order*, *Approve*, *Disapprove*, *Accept* and *Reject*. Some articles include a discussion of an issue, but the king's final decision is not explicitly recorded or the king postpones the decision. We treat this type of decision as *Discuss*, i.e., the sixth category. Finally, we choose fifteen kings with more than 200 articles that have his final decisions. Table 1 shows the basic statistics of our meeting records data from the AJD.

[1] http://sillok.history.go.kr/id/kda_10103027_005

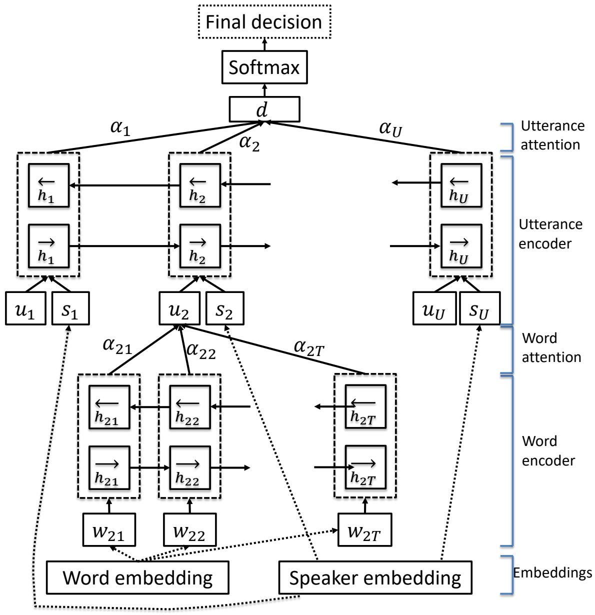


Figure 2: Conversational Decision-Making Model

3 Conversational Decision Making Model

This section describes our model, the Conversational Decision-Making Model (CDMM), for identifying leaders' decisions from meeting records. CDMM is based on the Hierarchical Attention Network (HAN) (Yang et al., 2016), but we change the sentence level to the utterance level and use speaker information (described in Section 3.2). To encode the speaker information, we build the speaker embedding from co-occurrence document network (described in Section 3.3).

3.1 Word Encoder

To encode the t-th word of i-th utterance x_{it}, $t \in \{1, \ldots, T\}$, we initially change the word x_{it} to word vector w_{it} using the word embedding matrix W_w, $w_{it} = W_w x_{it}$. We use a bi-directional GRU (Bahdanau et al., 2014), and concatenate the hidden states $h_{it} = [\overrightarrow{h_{it}}; \overleftarrow{h_{it}}]$. Then, we use the attention mechanism in HAN to find important words to classify the decision. Each word has an attention value α_{it}, and we compute the utterance word vector, $u_i = \sum_{t=1}^{T} \alpha_{it} h_{it}$.

3.2 Utterance Encoder with Speaker

In CDMM, the i-th utterance has word sequence representation vector u_i and speaker vector s_i. First, we change the speaker z_i to vector s_i using the speaker embedding matrix W_s, $s_i = W_s z_i$. To encode a length U of the utterances $(u_i, s_i), i \in \{1, \ldots, U\}$, we suggest encoders based on GRU (Bahdanau et al., 2014), which can learn u_i and s_i simultaneously, as follows:

$$
\begin{aligned}
h_i &= (1 - z_i) \odot h_{i-1} + z_i \odot \tilde{h}_i \\
z_i &= \sigma(W_{zu} u_i + W_{zs} s_i + U_z h_{i-1} + b_z) \\
r_i &= \sigma(W_{ru} u_i + W_{rs} s_i + U_r h_{i-1} + b_r) \\
\tilde{h}_i &= \tanh(W_{hu} u_i + W_{hs} s_i + r_i \odot (U_h h_{i-1}) + b_h)
\end{aligned}
$$

Here, h_i is the i-th utterance hidden state, and z_i and r_i denote the update and reset gate, respectively. This is similar to earlier work (Li et al., 2016), but we add the speaker vector to the utterance level, not the word level.

As in the word encoder, we use the bidirectional GRU with the utterance encoder and concatenate the hidden states $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$. We use the same attention mechanism to find important utterances. Each utterance has an attention value of α_i, and for the conversation vector we use $d = \sum_{i=1}^{U} \alpha_i h_i$.

With vector d, CDMM predicts the decision using softmax $p = softmax(W_c d + b_c)$, and a dropout scheme (Srivastava et al., 2014) to avoid over-fitting.

3.3 Pre-trained Speaker Embedding

Unlike word embedding which is pre-trained from news or Wikipedia articles (Mikolov et al., 2013; Bojanowski et al., 2017), pre-trained speaker embedding for the AJD does not exist. To overcome this limitation, we suggest the building of speaker embedding from the co-occurrence document network in the AJD. The AJD contains not only meeting records but also personnel management reports and explanations of the officials. We therefore build a co-occurrence network. The vertices are people, and two individuals are connected if they appear in the same article. The weight of the edge is the number of co-occurrences in the same article. With this network, we realize speaker embedding using the node2vec algorithm (Grover and Leskovec, 2016), which generates node vector representation.

4 Experiments

This section describes the experiments and results of CDMM as well as other methods for classifying the king's decisions in the AJD.

4.1 Experiment Setting

We split the data as 80/10/10 for training/validation/test. Because the meeting records contain fifteen kings, we split the data randomly for each king and merge each part into the entire training, validation and test set.

We compare CDMM with the following methods. The majority of classes predicts all test examples as the major class, *Discuss*. We apply Naive Bayes and the SVM with the linear kernel. To use these methods, we remove words whose document frequency is smaller than twenty. To see the power of the speaker information, we run these baselines on words and speaker features together. We also run fastText (Joulin et al., 2017), which is a classifier with n-gram features and hierarchical softmax, and is similar to CBOW (Mikolov et al., 2013). We use pre-trained Korean word vectors[2] (Grave et al., 2018) to fastText and CDMM. We create the speaker embedding from the AJD. For a fair comparison, we exclude the valid and test articles to construct the co-occurrence network. We use node2vec implementation[3] for speaker embedding. We set the GRU hidden state size to 200, the dimension of the speaker embedding to 200 and the dropout probability to 0.5 for CDMM.

4.2 Predictions of the King's Decision Results

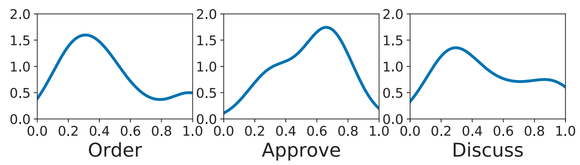
Table 2 shows the results. CDMM performs better than all other methods for macro-average and weighted-averaged metrics. The majority of classes shows the lowest performance. Naive Bayes and SVM outperform the baseline. fastText with pre-trained word vectors outperforms its counterpart, in accordance with an earlier result (Lample et al., 2016). CDMM without a speaker performs equally to HAN, the only difference being that HAN encodes sentences and CDMM encodes utterances. It does not show good performance as it models only the hierarchical structure of the conversation. However, when we add speaker information, the performance increases even with random initialization of speaker embedding. The performances of Naive Bayes and SVM also increase when they are assigned speakers as features. These observations signal that speaker information is helpful for predicting the king's decisions. Finally, CDMM with pre-trained speaker

[2] https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md

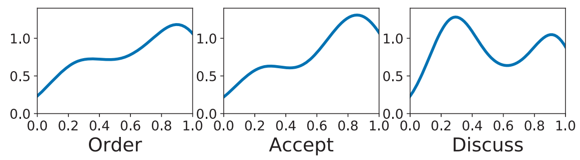
[3] http://snap.stanford.edu/node2vec

Method	*Micro* F_1	*Macro Prec*	*Macro Rec*	*Macro* F_1	*W-avg* F_1
Majority of classes	0.472	0.079	0.167	0.107	0.303
Naive Bayes	0.479	0.173	0.176	0.126	0.321
SVM linear	0.381	0.249	0.246	0.246	0.383
SVM RBF	0.487	0.236	0.186	0.142	0.337
Naive Bayes with speaker	0.466	0.268	0.177	0.135	0.323
SVM linear with speaker	0.423	0.292	0.259	0.243	0.403
SVM RBF with speaker	0.472	0.079	0.167	0.107	0.303
fastText w/o word vector	0.487	0.158	0.193	0.150	0.349
fastText	0.499	0.315	0.225	0.215	0.402
CDMM w/o speaker	0.481	0.176	0.214	0.178	0.379
CDMM with speaker (random init)	**0.504**	0.258	0.227	0.208	0.401
CDMM with speaker (pre-trained)	0.476	**0.329**	**0.307**	**0.313**	**0.456**

Table 2: King's decision classification precision, recall and F-measures. *Micro* $F1$ is the micro-averaged value of F-measure, and *Macro* $Prec$, Rec and F_1 are the macro-averaged values of precision, recall and F-measure respectively. *W-avg* F_1 is the weighted average according to the number of true examples in each class. CDMM outperforms all other methods compared.

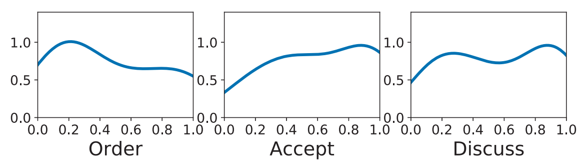


(a) Word "Wish to do"

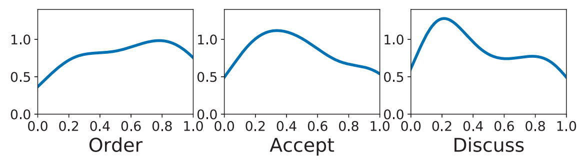


(b) Word "Okay"

Figure 3: Attention weight distribution of words for each class



(a) Word "Okay" from kings



(b) Word "Okay" from officials

Figure 4: Attention weight distribution of word for each class from kings and officials

embedding shows better results compared to all other methods.

5 Discussion

Here, we investigate the attention values to determine the important words and speakers for predicting the king's decisions. We also obtain evidence showing why CDMM with pre-trained speaker embedding outperforms the others.

5.1 Key Words and Speakers

We investigate the important words using word attention values. To find the important words, we compute the mutual information (Christopher et al., 2008) of words that have the top 10% of attention values in the utterances among the classes.

Figure 3 shows the attention weight distributions of the two examples of the top words "Wish to do" and "Okay". The word "Wish to do" is usually used to make a request to the king. The peak of the attention weight distribution of "Wish to do" for the *Approve* class is around 0.7, whereas it is around 0.3 for *Order* and *Discuss*. We can interpret this to mean that CDMM assigns greater attention to that word to predict *Approve* compared to *Order* and *Discuss*. The word "Okay" is used to consent to the opinions of others. CDMM assigns a high attention value to the word to predict *Order* and *Accept* compared to *Discuss*.

However, the attention values differ according to the speaker. As shown in Figure 4, CDMM gives a high attention score to the word "Okay" for

Name (Eng)	Position	Class
Sin Sukju	Secretary	*Order*
Jeong Changson	Secretary	*Order*
Kim Jonkyung	Local gov	*Approve*
Kim Neuk	Local gov	*Approve*
Gwon Jin	Local gov	*Disapprove*
Kim Seup	Remonstrator	*Disapprove*
Hwang Hui	Central gov	*Accept*
Han Myeonghoe	Central gov	*Accept*
Kim Jikyung	Remonstrator	*Reject*
Sung Damnyeon	Remonstrator	*Reject*

Table 3: Name (translated in English) and position of the speakers who have high mutual information scores for the classes. Local gov is the local government official and Central gov is the central government official. Remonstrator is the official who remonstrates to the king. The position of the speaker is important to predict the king's decision.

Accept as compared to the other classes when the speaker is king. However, when officials use this word, CDMM assigns a high attention value to the word in the *Order* class. Despite the fact that the same word is used, the king's decision is changed based on the speaker. This is additional evidence showing why the speaker information is useful to predict the decision.

5.2 Position of the Speaker

We investigate the key speakers from utterance attention values. To determine the important person, we use the same technique of finding important words.

We find that high ranking person's positions are shared for each class. Table 3 shows the top ranked speakers and their positions for each class. The chief secretary who takes orders from the king has a high rank in the *Order* class. For *Approve* and *Disapprove*, local authorities are highly ranked. For *Accept*, central government officials have high MI values. Interestingly, officials who remonstrate to the king have high scores in the *Disapprove* and *Reject* class. We can thus say that the kings refuse admonitions commonly from officials.

From these results, we can gain insight into why pre-trained speaker embedding is helpful to predict the king's decisions. People in the same organization are in the same community of co-occurrence news article network (Özgür et al., 2008). Therefore, the AJD network contains the community information, and node2vec generates the node's closeness via embedding. CDMM can have this knowledge in the model therefore outperforms the other methods.

6 Conclusion

In this paper, we created conversational meeting data from the Annals of the Joseon Dynasty (AJD). We presented Conversational Decision-Making Model (CDMM) to predict leaders' decisions from the data. We also suggested the use of speaker embedding from co-occurrence document network with node2vec. With this data, we showed that CDMM outperforms other methods in terms of most metrics. We implemented CDMM using tensorflow (Abadi et al., 2016), and published the code and data in public[4]. We also analyzed the reasoning behind the success of CDMM and the key words and speakers by investigating the concept of attention.

Studies of small group dynamics can be helpful when attempting to understand group decision making behavior (Backstrom et al., 2006). Prior work which analyzed small group dynamics relied on a hidden Markov model (Magdon-Ismail et al., 2003), a dynamic Bayesian network (Mathur et al., 2012) or a layered probabilistic model (Cheng et al., 2014) for various datasets such as networks or recorded video. We suggest CDMM, which combine two types of data to predict leaders' decision. We can also apply this idea to other group dynamics analyses.

Acknowledgments

We would like to thank Jooyeon Kim for the helpful discussion and the anonymous reviewers for the inspiring questions and comments. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-01778, Development of Explainable Human-level Deep Machine Learning Inference Framework).

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard,

[4]https://github.com/NoSyu/CDMM

et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI*, pages 265–283.

Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. 2006. Group formation in large social networks: Membership, growth, and evolution. In *Proceedings of the SIGKDD*, pages 44–54. ACM.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Trung H Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In *Proceedings of the SIGDIAL*, pages 235–243.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 28–39. Springer.

Zhongwei Cheng, Lei Qin, Qingming Huang, Shuicheng Yan, and Qi Tian. 2014. Recognizing human group action by layered model with multiple cues. *Neurocomputing*, 136:124–135.

D Manning Christopher, Raghavan Prabhakar, and Schacetzel Hinrich. 2008. Introduction to information retrieval. *An Introduction To Information Retrieval*, 151(177):5.

Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters. 2008. Modelling and detecting decisions in multi-party dialogue. In *Proceedings of the SIGDIAL*, pages 156–163.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the LREC*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the SIGKDD*, pages 855–864. ACM.

Pei-Yun Hsueh and Johanna D Moore. 2007. What decisions have you made?: Automatic decision detection in meeting conversations. In *Proceedings of the NAACL HLT*, pages 25–32.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the EACL*, pages 427–431.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the NAACL HLT*, pages 260–270.

James R Larson Jr, Pennie G Foster-Fishman, and Timothy M Franz. 1998. Leadership style and the discussion of shared and unshared information in decision-making groups. *Personality and Social Psychology Bulletin*, 24(5):482–495.

Kurt Lewin, Ronald Lippitt, and Ralph K White. 1939. Patterns of aggressive behavior in experimentally created “social climates”. *The Journal of social psychology*, 10(2):269–299.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the ACL*, pages 994–1003.

Frank C Lunenburg. 2011. Decision making in organizations. *International journal of management, business, and administration*, 15(1):1–9.

Malik Magdon-Ismail, Mark Goldberg, William Wallace, and David Siebecker. 2003. Locating hidden groups in communication networks using hidden markov models. In *International Conference on Intelligence and Security Informatics*, pages 126–137. Springer.

Shobhit Mathur, Marshall Scott Poole, Feniosky Pena-Mora, Mark Hasegawa-Johnson, and Noshir Contractor. 2012. Detecting interaction links in a collaborating group using manually annotated data. *Social Networks*, 34(4):515–526.

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

Arzucan Özgür, Burak Cetin, and Haluk Bingol. 2008. Co-occurrence network of reuters news. *International Journal of Modern Physics C*, 19(05):689–702.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Victor H Vroom and Arthur G Jago. 1988. *The new leadership: Managing participation in organizations.* Prentice-Hall, Inc.

Maria C Yang. 2010. Consensus and single leader decision-making in teams using structured design methods. *Design Studies*, 31(4):345–362.

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 NAACL: HLT*, pages 1480–1489.