언어학박사 학위논문

Fusion models for news quality prediction:

Combining textual features with sentence embeddings

뉴스 품질 예측을 위한 혼합 모형

— 텍스트 자질과 문장 임베딩 —

2021년 8월

서울대학교 대학원
언어학과 언어학전공
박 수 지

Fusion models for news quality prediction:
Combining textual features with sentence embeddings

뉴스 품질 예측을 위한 혼합 모형
— 텍스트 자질과 문장 임베딩 —

지도교수 신 효 필

이 논문을 언어학박사 학위논문으로 제출함

2021년 7월

서울대학교 대학원
언어학과 언어학전공
박 수 지

박수지의 박사학위논문을 인준함

2021년 8월

위 원 장:	남 승 호
부위원장:	신 효 필
위　　원:	강 승 식
위　　원:	최 수 진
위　　원:	김 문 형

Abstract

Fusion models for news quality prediction:

Combining textual features with sentence embeddings

PARK Suzi
Department of Linguistics
The Graduate School
Seoul National University

This paper aims to develop a language model to predict the quality of Korean news articles. The task of predicting the quality of news articles is that the latest techniques for natural language processing have yet to be applied, even though the need has emerged due to the recent flood of fake news. To overcome these limitations, we develop an SBERT (Sentence BERT) model that represents the meaning of a sentence to examine whether the performance of quality classification can be enhanced by utilizing the linguistic features of the article. As a result, both machine learning models using textual features such as readability and cohesion in articles and transfer learning models using contextual features automatically extracted from SBERT performed better than previous studies, specifically when augmenting and refining training data in

SBERT learning. Thus, we conclude that linguistic features play an essential role in the quality of the article and that SBERT, a state-of-the-art technique for natural language processing, can contribute to the extraction and utilization of linguistic features.

Keywords: Computational linguistics, Sentence embeddings, News quality prediction, Fusion models, SBERT

Student Number: 2015-30035

Contents

List of Tables

List of Figures

1 Introduction

In the past decade, various researchers have found it necessary to develop a system to predict article quality automatically. Many of these studies have shown remarkable results using traditional machine learning algorithms such as support vector machines and manual feature engineering.

Distinguishing high- from low-quality news is crucial for preventing misinformation and curating information. Although substantial research has tackled this challenging question, there is still a gap for the most recent advances in natural language processing (NLP).

This study aims to develop a model to predict news articles' quality level evaluated by readers automatically, exploiting linguistic features from a news article text and using a Transformer language model. With this aim, we set our research objectives as follows:

- To identify which manually engineered textual features affect audience-rated news quality.
- To develop sentence representation models to utilize effects of automatically extracted contextual features.
- To combine two types of linguistic features and to improve performance in predicting news quality.

Then this thesis explores the following research questions:

1. What textual features affect news quality?

2. Are sentence representation models more effective for long articles than word representation models?

3. Does combining two models improve performance in predicting news quality?

The first contribution of this thesis is the development of a news quality prediction system applicable to new articles by only use linguistic features without implementing a further news survey. It is important that no further survey is required because the amount of articles published at every moment is enormous, and it is impossible to collect information on all of them manually. On the other hand, linguistic features can be calculated directly from text only, so they are inexpensive and easy to apply.

The second contribution is the development of sentence representation models for the Korean language. We demonstrate that these models effectively process long documents with multiple sentences through the article quality prediction task.

Despite the improvement that we achieved with the proposed scheme, our models' performance did not surpass the best results of existing work with extralinguistic features such as journalistic values and demographic information. However, we remark that the efficiency of language features can compensate for this limitation. We also believe that we can improve the performance of our model by reflecting linguistic characteristics of articles more broadly in the future.

In Chapter 1, we have introduced the context of the study, identified the research questions, argued the value of such research. In addition, we have discussed the limitations of the study.

Chapter 2 will review the existing literature to identify approaches to news quality prediction within the context of natural language processing.

Chapter 3 will describe the data[1] that we use and present the method that we adopted. We will justify focusing on writing styles other than journalistic values and demographical information. Subsequently, we will discuss the research design for exploiting linguistic features from the news article text.

Chapter 4 will explore what linguistic cues make a news article seem superior to others. First, we feed manually engineered textual features into an ordinal logistic regression model. Second, we investigate which factors significantly affect news quality. Finally, we find three interesting facts about high-quality articles.

Chapter 5 will focus on automatically extracted contextual features. For this purpose, we develop KR-SBERT, a Korean Sentence Transformer model representing the meaning of a sentence. Then we evaluate our new models in the task of news quality prediction and compare them with BERT.

Chapter 6 will present our novel approach. We maximize the effects of two kinds of linguistic features that we have built in the previous two chapters

[1] In this thesis, we use the dataset from the following paper with the courtesy of its first author:

- Choi, S., Shin, H., & Kang, S. S. (2021). Predicting Audience-Rated News Quality: Using Survey, Text Mining, and Neural Network Methods. *Digital Journalism, 9(1)*: 84–105.

The linguistic attributes used in the paper are overlapping with our textual features in Chapter 4. When working as a research assistant for the above project from 2018 to 2019, the author of this thesis performed morphological analysis, defined a list of morphemes for linguistic features, and wrote algorithms and codes to extract the features from articles.

by fusing the textual feature model and the contextual feature model in two methods. Experimental results show that our approach is successful.

Chapter 7 concludes the preceding chapters, summarizes the problems that we solved, and addresses this study's limitations and directions for future research.

2 Literature Review

This chapter outlines what has been discussed in previous research on predicting news articles' quality throughout four sections; First, it deals with the historical background of news quality prediction. Second, it identifies data, features, and models that researchers use in natural language processing (NLP) for this task. Third, it introduces the recent studies on instruments and techniques that we will use in this thesis. Then, we will determine what insights we can gain and what contributions we can make in this literature.

2.1 Background

This section explores two research lines related to our work: (i) text classification and (ii) news quality assessment. Since news article quality prediction is a subtask of text classification, we begin with its definition and history. Then, we review how to assess news quality before predicting it.

2.1.1 Text Classification

2.1.1.1 Initial Studies

Text Classification, or *Text Categorization*, is the task of automatically assigning documents to a predefined set of categories (Foltz, 1990; Foltz et al., 1998; Joachims, 1998; Sebastiani, 2002). Its history dates back to the 1960s. Maron (1961) first defined the task of "classifying *linguistic* entities" for *Information Retrieval* and Borko and Bernick (1963) generalized his work experimentally to prove that automatic document classification is a possible task. Studies that were developed in the 1970s continued this tradition and tended to use the occurrence and frequency of keywords in documents as predictors (Heaps, 1973; Kar, 1975; White et al., 1975, 1977; Hamill and Zamora, 1980).

As Schütze et al. (1995) and Marton et al. (2005) pointed out, with the introduction of machine learning since the 1990s, approaches to text categorization (Lewis and Ringuette, 1994) have used classification trees (Tong and Appelbaum, 1994; Lewis, 1992a), Bayesian classifiers (Lewis, 1992a,b; Peng et al., 2004), rules induction (Apte et al., 1994), nearest-neighbor techniques (Masand et al., 1992; Yang, 1994), neural networks (Tong and Appelbaum, 1994), and logistic regression (Zhang et al., 2003).

2.1.1.2 News Classification

Until the 1980s, the main subjects of text classification studies were short texts, including abstracts from computer science (Maron, 1961; Borko and Bernick, 1963), physics (Kar, 1975; Biebricher et al., 1988), and chemistry (Hamill and Zamora, 1980), and telegraphic messages Young and Hayes (1985); Goodman (1990).

In the late 1980s, Hayes et al. (1988) attempted to classify longer texts, such as news stories. Then, Hayes and Weinstein (1990) extended their pilot study to develop Construe, a rule-based news categorization system. As shown in Figure 2.1, its main processing steps are concept recognition and categorization rules. The rule developers defined concepts as patterns of words and phrases in context and controlled decisions using if-then rules and boolean combinations. Meanwhile, Rau and Jacobs (1991) built a news categoriza-

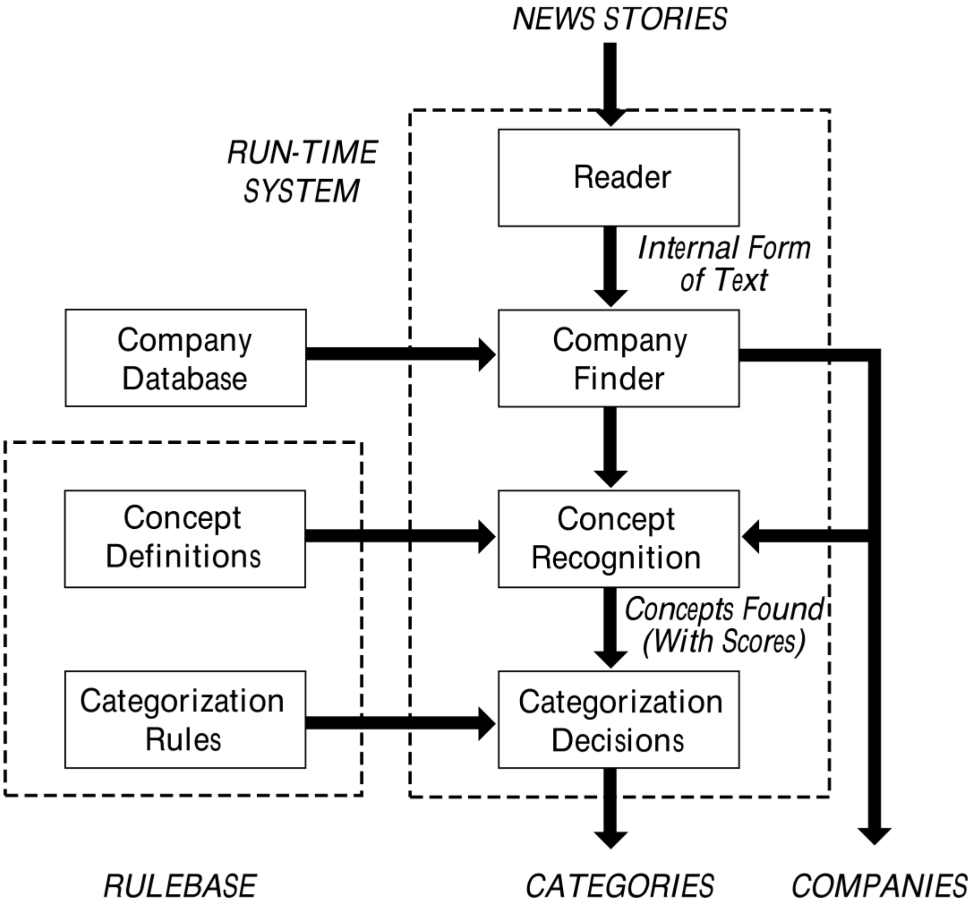


Figure 2.1: Construe's flow of control (Hayes and Weinstein, 1990)

tion system named NLDB and improve retrieval accuracy using Segmented Databases, Text Category Browsing, Query by Relationship, and Special Name Handling.

While most of the research at this time relied on numerous rules and an extensive database, Masand et al. (1992) classified news stories using a k-nearest neighbor method, not requir ing manual topic definition. Since the 1990s, public news classification data such as 20 Newsgroup Dataset (Rennie, 2005a,b) became accessible. Research using machine learning techniques, including Naive Bayes (NB) (Danesh et al., 2007) classifiers and Support Vector Machine (SVM) (Sun et al., 2009; Kumar and Gopal, 2010), evolved.

Recent news classification studies cover various subtasks, including news recommendations (Chiang and Chen, 2004; Bogers and van den Bosch, 2007; Cantador et al., 2008; Wang et al., 2010a,b; An et al., 2019; Wu et al., 2019, 2020; Hu et al., 2020), fake news detection (Rubin et al., 2016; Rashkin et al., 2017; Bourgonje et al., 2017; Thorne et al., 2017; Karimi et al., 2018), and news quality prediction.

2.1.2 Text Quality Assessment

To predict classify news quality, we should first determine how to assess it. NLP researchers presented readability and coherence as criteria and proposed methods for computing them using lexical, syntactic, and discourse properties to assess text quality (Louis, 2012, 2013). Mesgar and Strube (2018) developed a neural local coherence model to capture the coherence between adjacent sentences and showed that it is beneficial for news readability assessment.

There have also been studies to identify factors that degrade news quality, such as propaganda and extremity. For example, Da San Martino et al. (2019) performed a fine-grained analysis of news texts to detect propaganda.

On the other hand, recent journalism studies have consistently attempted to measure news quality from users' perspectives (Zaller, 2003; Urban and Schweiger, 2014; Costera Meijer and Bijleveld, 2016; Maddalena et al., 2018; Molyneux and Coddington, 2020; Bachmann et al., 2021). As a result, they are showing that audiences are capable of evaluating news quality.

2.2 News Quality Prediction Task

As a subtask of document classification, news quality prediction assigns a given news article to a quality label, which human annotators predefined. In this section, we cover data types and methodology that studies on this task have used. A synopsis of literature relating to article quality prediction is reported in Table 2.1.

2.2.1 News Data

2.2.1.1 Online vs. Offline

Over the past decade, NLP researchers have focused on determining the quality of articles and predicting them automatically. Their subjects include both newspapers and online news. For example, Louis and Nenkova (2013) first studied article quality prediction in the science journalism domain, Ferschke (2014), Dang and Ignat (2016) and Guda et al. (2020) assessed the quality of Wikipedia articles, and Arapakis et al. (2016) and Samarinas and Zafeiriou (2019) quantified the quality of online news.

2.2.1.2 Expert-rated vs. User-rated

The criteria for defining the quality level of articles were largely expert-centered and user-centered. As an example of the former, Louis and Nenkova (2013) set two categories, VERY GOOD and TYPICAL, and classified an article as VERY

Year	**Author(s)**	**Data domain**	**Quality**	**Evaluation criteria**	**Features**	**Classifier**
2013	Louis and Nenkova	Science journal	·VERY GOOD ·TYPICAL	Whether the author appears in “The Best American Science Writing” anthology	41 dimensions ·readability ·well-written nature ·interesting fiction ·content	SVM
2016	Dang and Ignat	Wikipedia	·SA ·GA ·B ·C ·START ·STUB	Assigned by the assessment department of the Years WikiProject	Extracted from Doc2Vec	DNN
2016	Arapakis et al.	Online news	Editorial quality	Assessed by ten expert judges who had a background in computational linguistics, journalism, or were media monitoring experts	14 dimensions ·readability ·informativeness ·style ·topic ·sentiment	GLM
2017	Volkova et al.	Twitter news posts	·SUSPICIOUS ·VERIFIED	Relied on public resources that annotate suspicious Twitter accounts	·Bias cues ·Subjectivity cues ·Psycholinguistic cues ·Moral foundation cues	LSTM, CNN
2019	Samarinas and Zafeiriou	Online news	·LOW (click-bait) ·HIGH (non click-bait)	Collected from a pseudo-news website and Reuters	Extracted from Fast-Text pre-trained embeddings	BiLSTM

Table 2.1: Synopsis of article qualiy prediction literature

GOOD if its author appears in “The Best American Science Writing” anthology, which science experts edit. In addition, Arapakis et al. (2016) used input from news editors, journalists, and computational linguists. In the studies above, it was professionals that judged the quality of articles.

In this thesis, we follow Choi et al. (2021) in using quality levels evaluated by audiences for newspaper articles in their survey. In the following two subsections, we explore the background for setting factors and determining models.

2.2.2 Prediction Methods

2.2.2.1 Manually Engineered Features v. Automatically Extracted Features

In order to automatically predict the quality of newspaper articles, it is essential to capture and reflect the characteristics of the text that forms newspaper articles. The linguistic charactaristics of text can be classified into several categories. Louis and Nenkova (2013) used 41 features on readability, well-written nature, interesting fiction, and content. Ferschke (2014) defined linguistic quality in terms of Language correctness, writing traints and rubrics, readability, and text organization. Arapakis et al. (2016) identified 14 content aspects on readability, informativeness, style, topic, and sentiment. Generally, these stylistic features are defined by hand-crafted rules, and traditional linguistic tools such as n-grams are used to obtain values for these features.

In contrast, features can be obtained by automatic extraction, particularly in deep neural networks. Dang and Ignat (2016) proposed an approach that uses Doc2Vec (Le and Mikolov, 2014) for learning features from textual documents. Guda et al. (2020) aggregated contextual features automatically using BERT (Devlin et al., 2019).

To the best of our knowledge, no research has yet been done in the article quality classification task considering both manual and automatic features. Therefore, we will consider both of these features in our study and finally combine them in a fusion model. In addition, we will also leverage SBERT to obtain contextual features.

2.2.2.2 Machine Learning vs. Deep Learning

Research on news quality prediction has been first conducted with traditional machine learning algorithms, such as support vector machines (SVMs) (Louis and Nenkova, 2013) or regression models (Arapakis et al., 2016). More recently, on the contrary, studies started to adopt deep neural networks (Dang and Ignat, 2016), Bidirectional Long Short-Term Memory (BiLSTM) networks, (Volkova et al., 2017), and Convolutional Neural Networks (CNN) (Samarinas and Zafeiriou, 2019).

There have been attempts to combine machine learning and deep learning methods in several subfields of NLP. For example, as shown in Figure 2.2, Alhindi et al. (2020) combined SVM, recurrent neural networks (RNNs), and BERT to classify news and editorials. Similarly, Figure 2.3 shows how Cao et al. (2020) incorporated BiLSTM and BERT to diagnose grammatical errors. However, this thesis is the first to fuse two models for predicting news quality as far as we know.

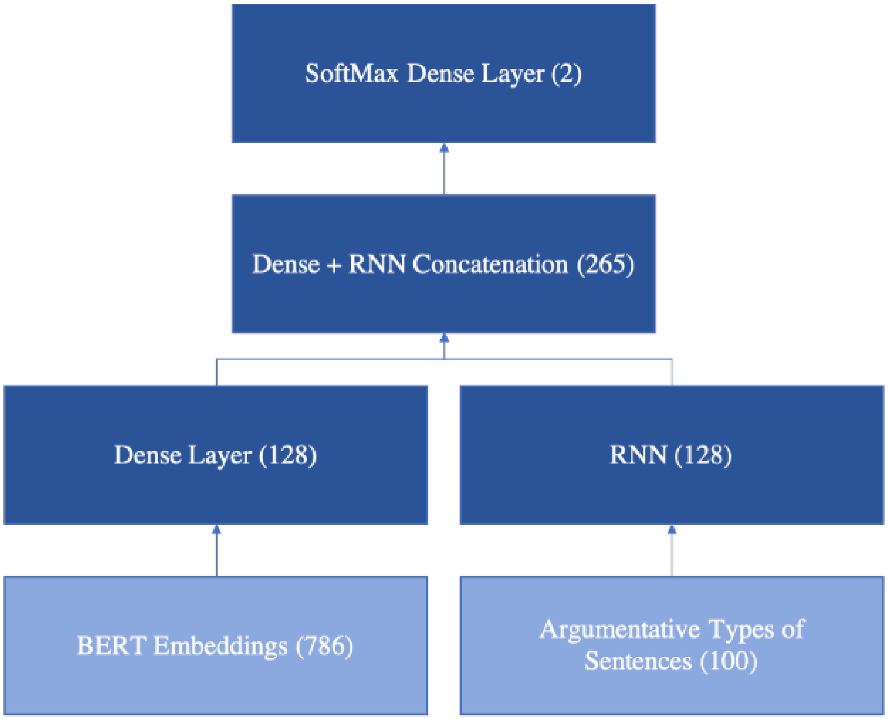


Figure 2.2: RNN + BERT architecture (Alhindi et al., 2020)

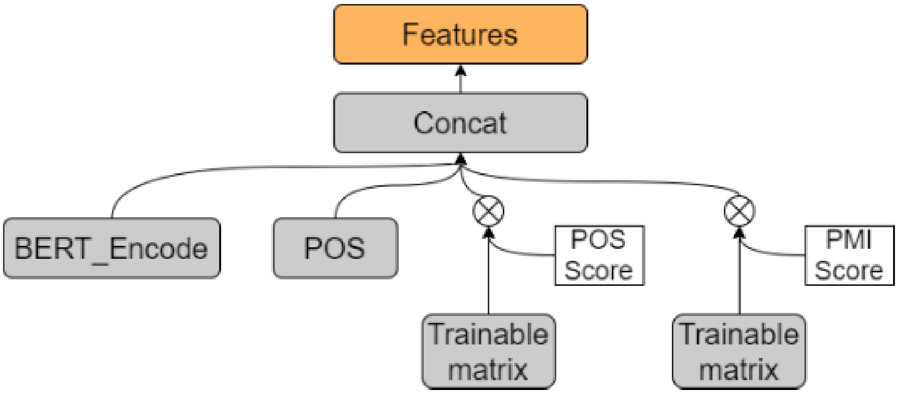


Figure 2.3: BERT with Score-feature Gates (Cao et al., 2020)

2.3 Instruments and Techniques

This section presents the state-of-the-art NLP tools that we will use in this thesis. The first is a model for representing the meaning of news articles, and the second is a technique for integrating results from multiple models of the same data.

2.3.1 Sentence and Document Embeddings

The synopsis in Table 2.1 illustrated the features used to predict news quality, including manually engineered and automatically extracted. The latter is extracted from embedding models, such as FastText and Doc2Vec, to capture semantic properties of words and sentences in a news article. These models represent a word, sentence, or paragraph as an element of a vector space so that vectors from semantically similar words get similar in the vector space. This section describes Word2Vec and its paragraph version, Doc2Vec, as the most widely used embedding technique and introduces BERT and SBERT as recent contextual embedding techniques.

2.3.1.1 Static Embeddings

Word2vec (Mikolov et al., 2013a,b) is a language model consisting of the continuous-bag-of-words (CBOW) and skip-gram (SG) architectures. As shown in Figure 2.4, the CBOW model predicts a single target word from its surrounding words, and the SG model uses a target word as input to obtain its context words as output. Each model trains shallow neural networks to learn the weights of words, which comprise a vector representing the target word.

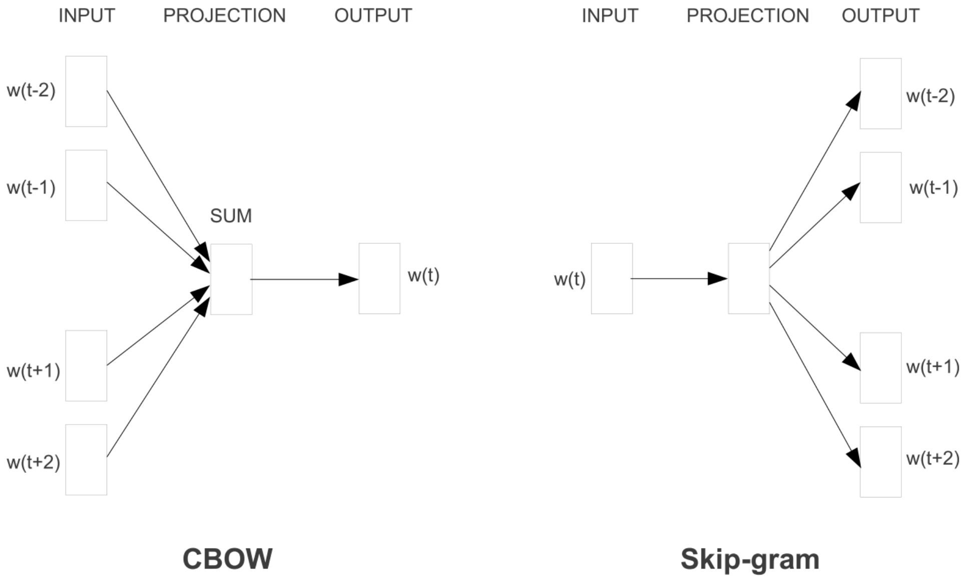


Figure 2.4: Two architectures of Word2Vec. The CBOW architecture predicts the current word based on thecontext, and the Skip-gram predicts surrounding words given the current word. (Mikolov et al., 2013b)

This vector captures the semantic properties of the word. Word2vec is a *static* embedding method because it assigns an invariant vector to a word type.

Doc2Vec Like Word2Vec, the Doc2Vec (Le and Mikolov, 2014) algorithm trains a dense vector to predict words in a given context. However, unlike Word2vec, its vectors represent a sentence, paragraph, or document consisting of a variable number of words. In its Paragraph Vectors' Distributed Memory (PV-DM) model (see Figure 2.5), the paragraph vector is concatenated into word vectors and learned to predict the next word. In another model, Distributed Bag of Words (PV-DBOW), the paragraph vector is trained to predict the words in a small window.

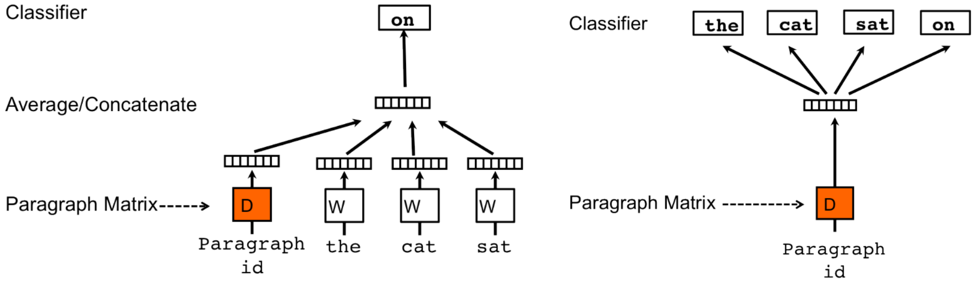


Figure 2.5: Doc2Vec frameworks for learning paragraph vector (Le and Mikolov, 2014)

2.3.1.2 Contextual Embeddings

Contextual embeddings have the advantage of capturing a word's meaning that changes across multiple contexts. For example, they can learn two different meanings of *baked* in the following sentences.

(1) John *baked* the potato.

(2) John *baked* the cake.

The Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) model consists of multiple layers of bidirectional transformers mapping input embeddings to contextual embeddings (see Figure 2.6). A sentence is segmented into subword tokens in BERT's architecture, and each token has token, segment, and position embeddings as in Figure 2.7. Their sums, as input embeddings, are fed into the multi-layer bidirectional transformer and transformed into contextual embeddings. BERT is pre-trained on two unsupervised tasks: masked language modeling and next sentence prediction.

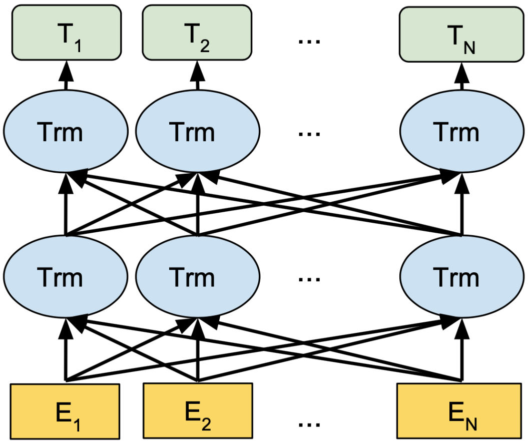


Figure 2.6: BERT pre-training architecture: Bidirectional Transformer (Devlin et al., 2019)

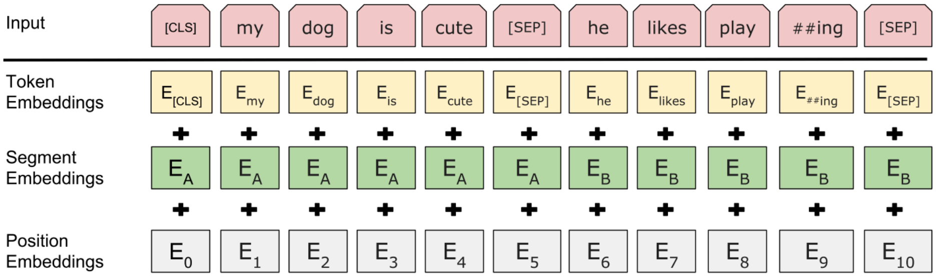


Figure 2.7: BERT input representation (Devlin et al., 2019)

SBERT Sentence-BERT, or SBERT, is a modification of the BERT network. Its network is called siamese because two sentences are encoded using the same transformer model. The SBERT architecture encodes each sentence into a sequence of contextualized token vectors using the BERT network and pools the token vectors into a single sentence vector. Then, it fine-tunes its siamese network, illustrated in Figures 2.8–2.9, by updating the weights "such that the produced sentence embeddings are semantically meaningful and can be compared with cosine-similarity."

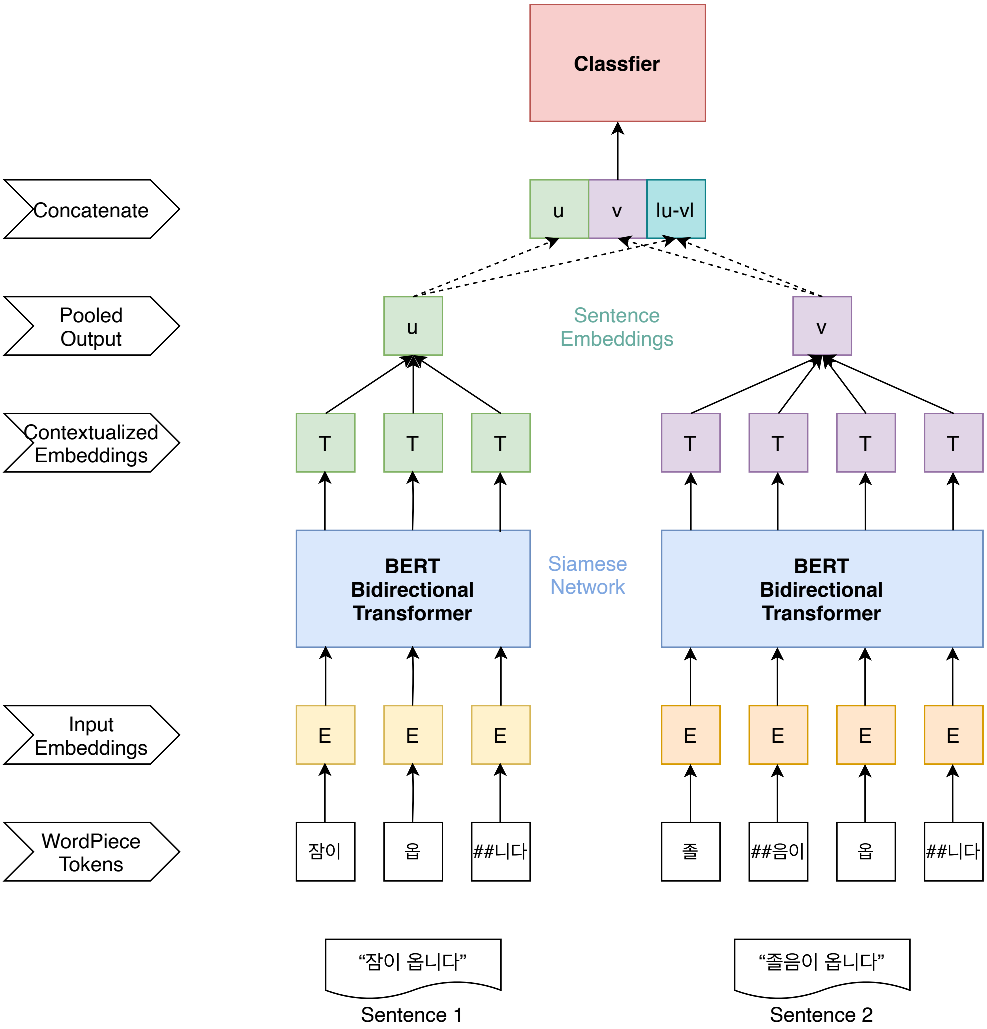


Figure 2.8: SBERT arcitecture at classification using siamese networks

2.3.2 Fusion Models

Finally, to combine SBERT models with other existing models, we examine the methods of applying multiple models to a single task. From feature extraction to class decision, we can fuse models at various levels (Aygunes et al., 2021).

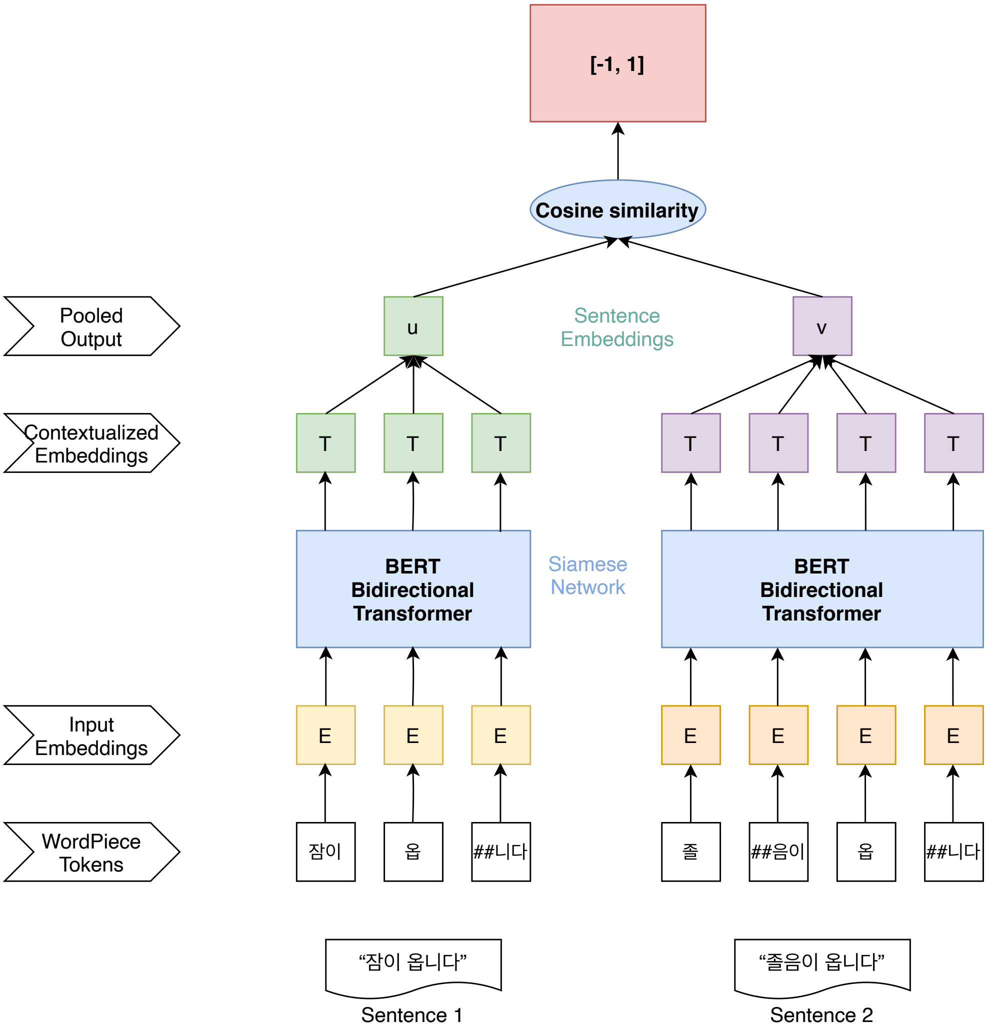


Figure 2.9: SBERT arcitecture at inference using siamese networks

Feature-level Fusion First of all, the feature-level fusion is the shallowest unless multimodality is assumed. Two or more feature vectors are concatenated and then fed into a classifier or a regressor at this level. The subsequent process is the same as a single model (see Figure 2.10).

Many studies use this fusion method (Alhindi et al., 2020; Cao et al., 2020;

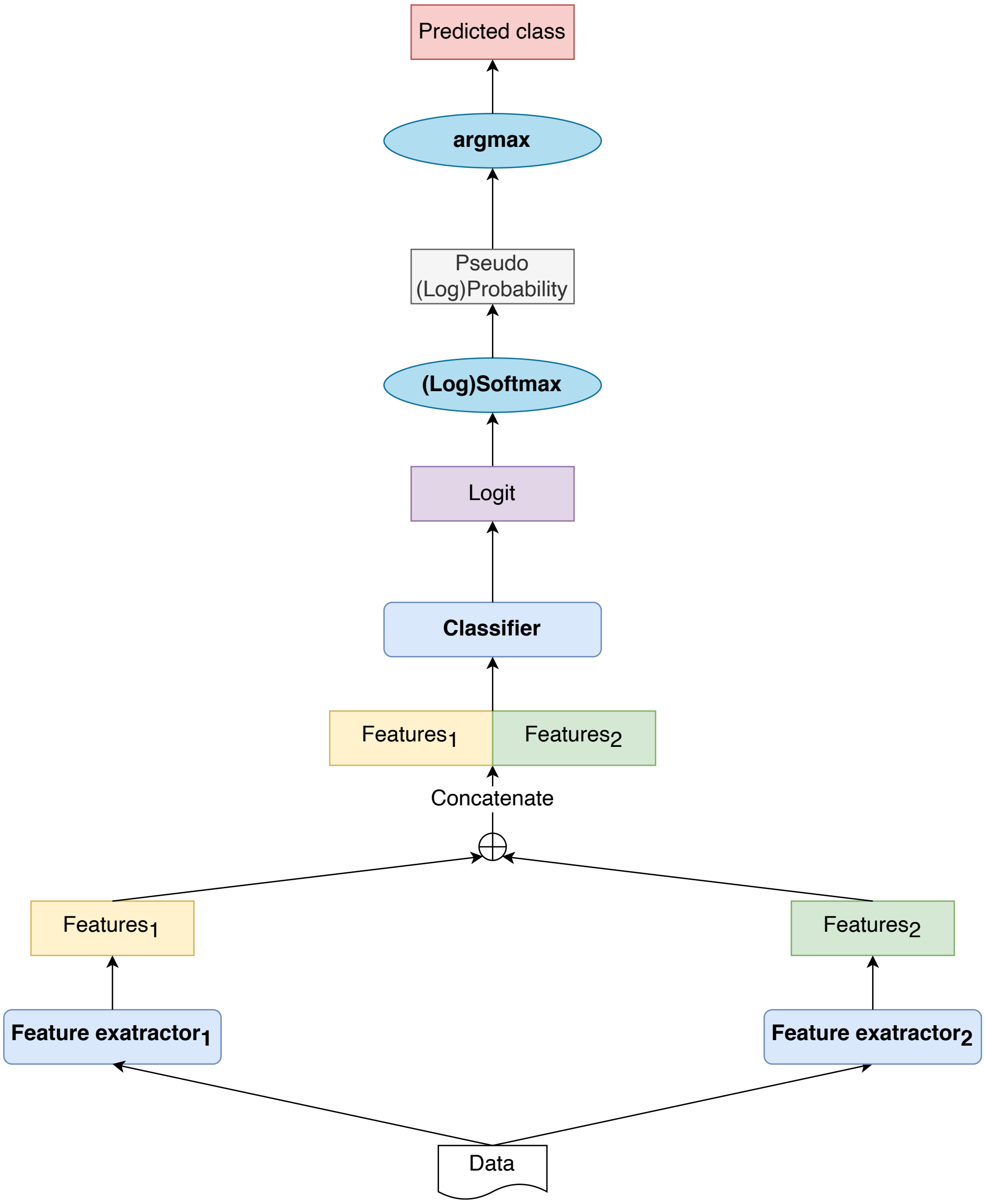


Figure 2.10: Feature-level model fusion

Rezvani et al., 2020; Wang et al., 2021). For example, a document can have both hand-crafted features and contextualized features.

Logit-level Fusion From this level, we can deal with multiple models. At this level, we keep two classifiers separately but fuse their logit values by calculating their sum (see Figure 2.11). Then we use the fused logit to get the losses.

Probability-level Fusion The probability-level fusion (Alkoot and Kittler, 2000) is similar to the logit-level, but two models are combined after a softmax layer (see Figure 2.12). This method is appropriate for models not using softmax cross-entropy losses.

Loss-level Fusion This method is also called External Fusing (Lee, 2021). It is appropriate for binary classification. For multiclass classification, it is difficult to reflect the models‘ effects on the probability of each class.

Decision-level Fusion In this method, a majority vote determines the classification results (Vildjiounaite et al., 2009; Ali and Ragb, 2019; Rezvani et al., 2020) (see Figure 2.13). It needs more than two models.

Due to the limitations described above, we will try two fusion methods in this thesis: feature-level and logit-level.

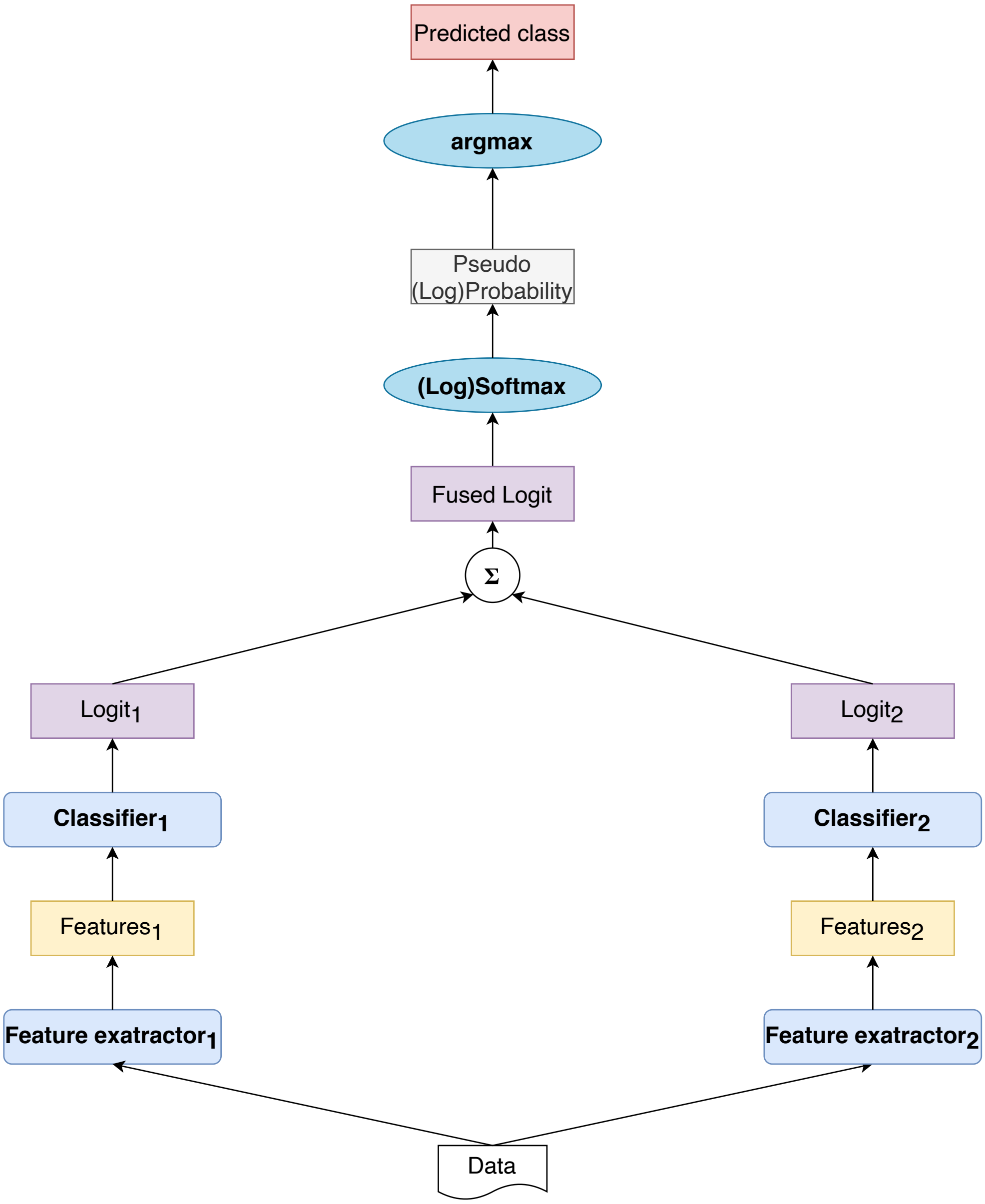


Figure 2.11: Logit-level model fusion

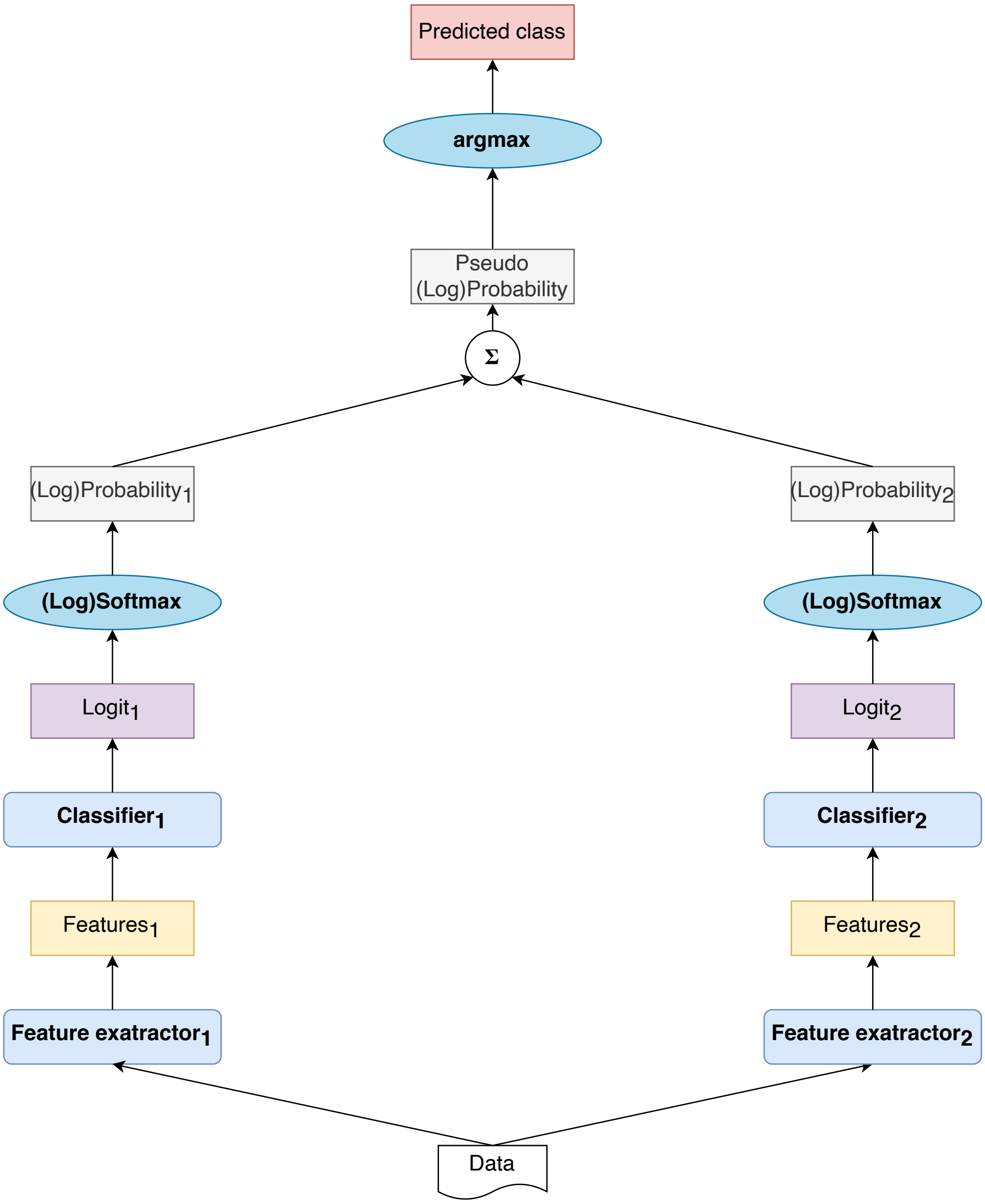


Figure 2.12: Probability-level model fusion

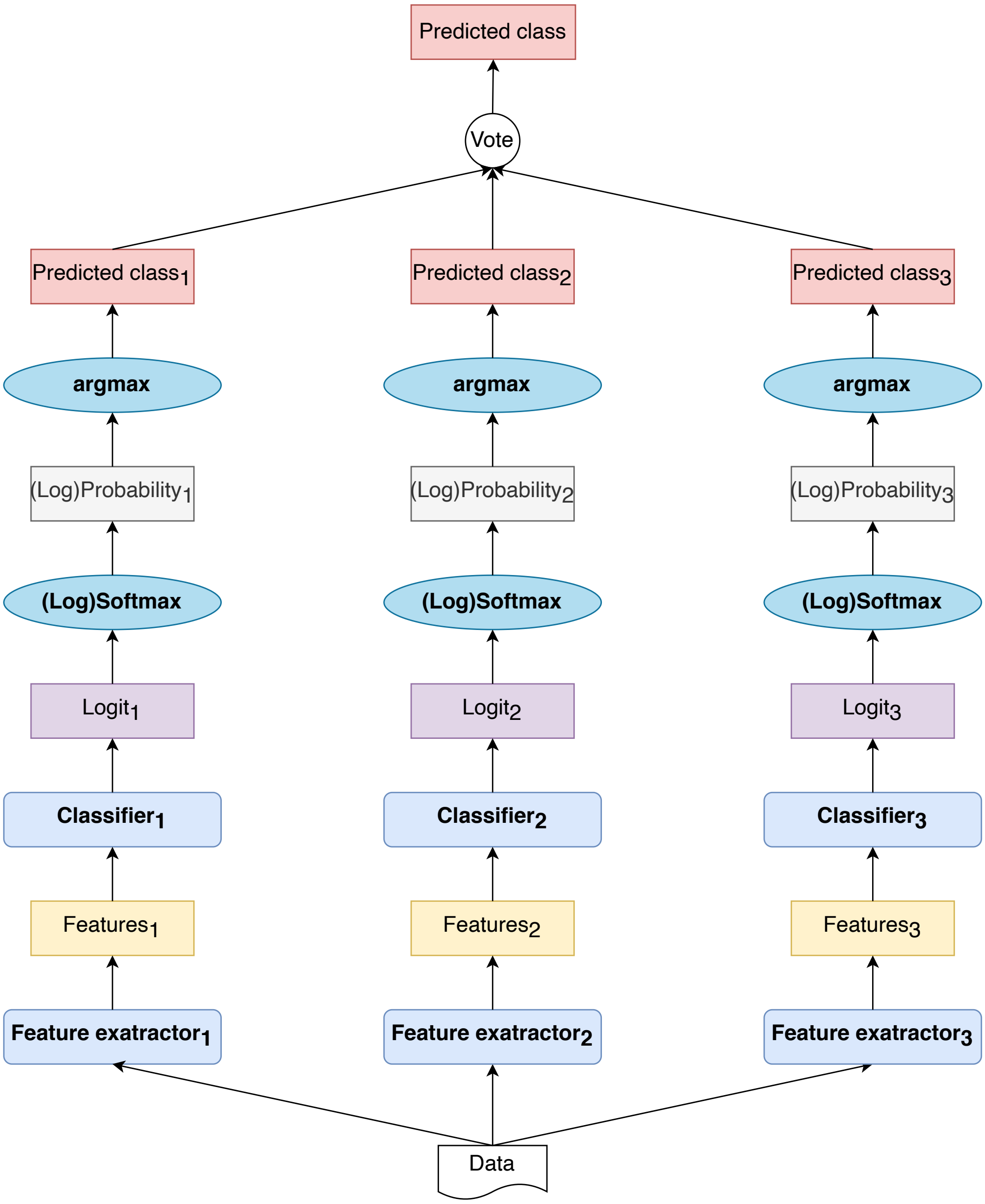


Figure 2.13: Decision-level model fusion

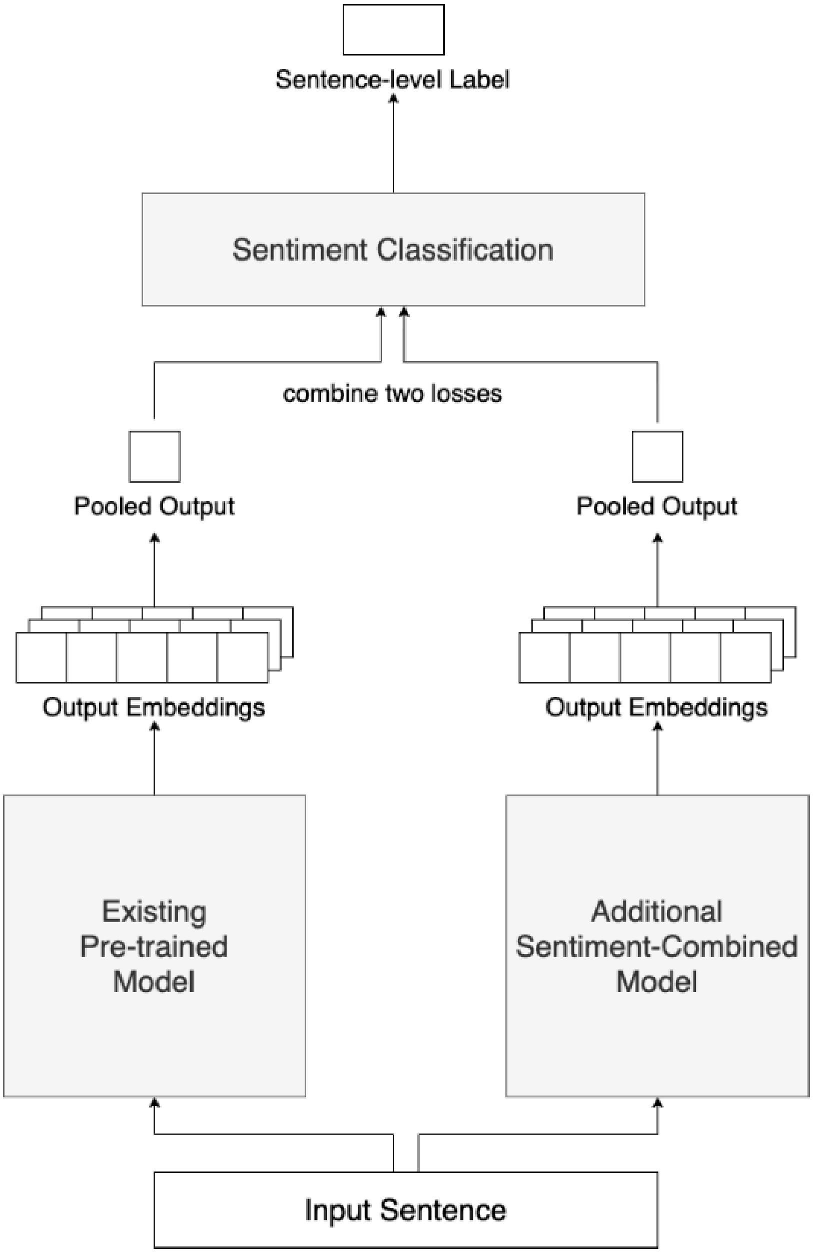


Figure 2.14: External fusing methods (Lee, 2021)

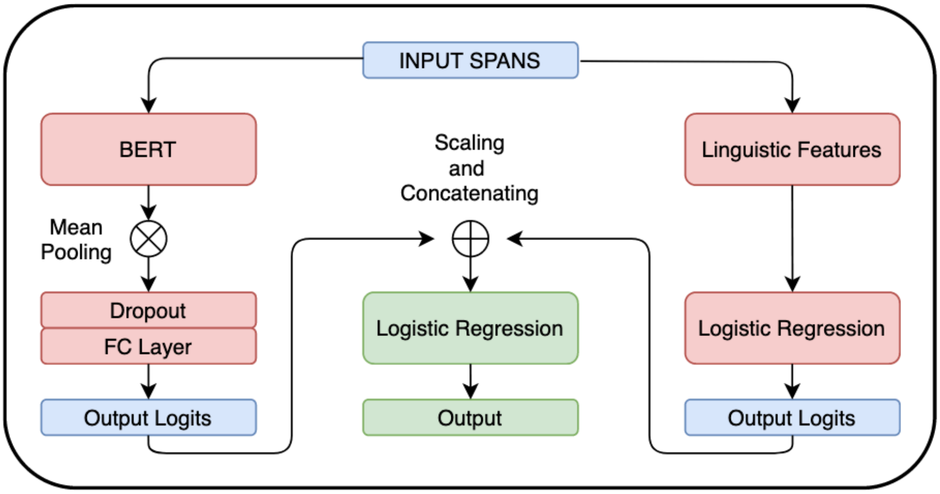


Figure 2.15: Ensemble Model Architecture for Technique Classification (Patil et al., 2020)

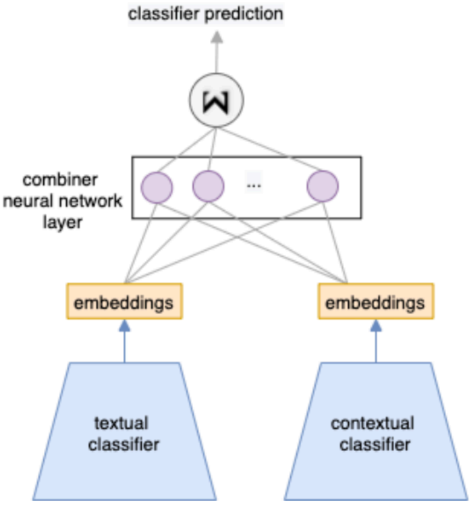


Figure 2.16: Neural network combining textual and contextual features (Rezvani et al., 2020)

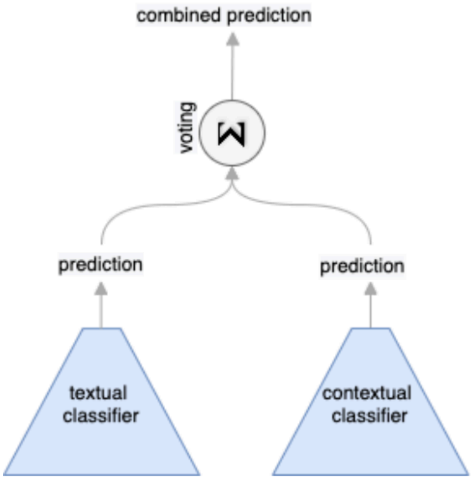


Figure 2.17: Booster model combining textual and contextual features (Rezvani et al., 2020)

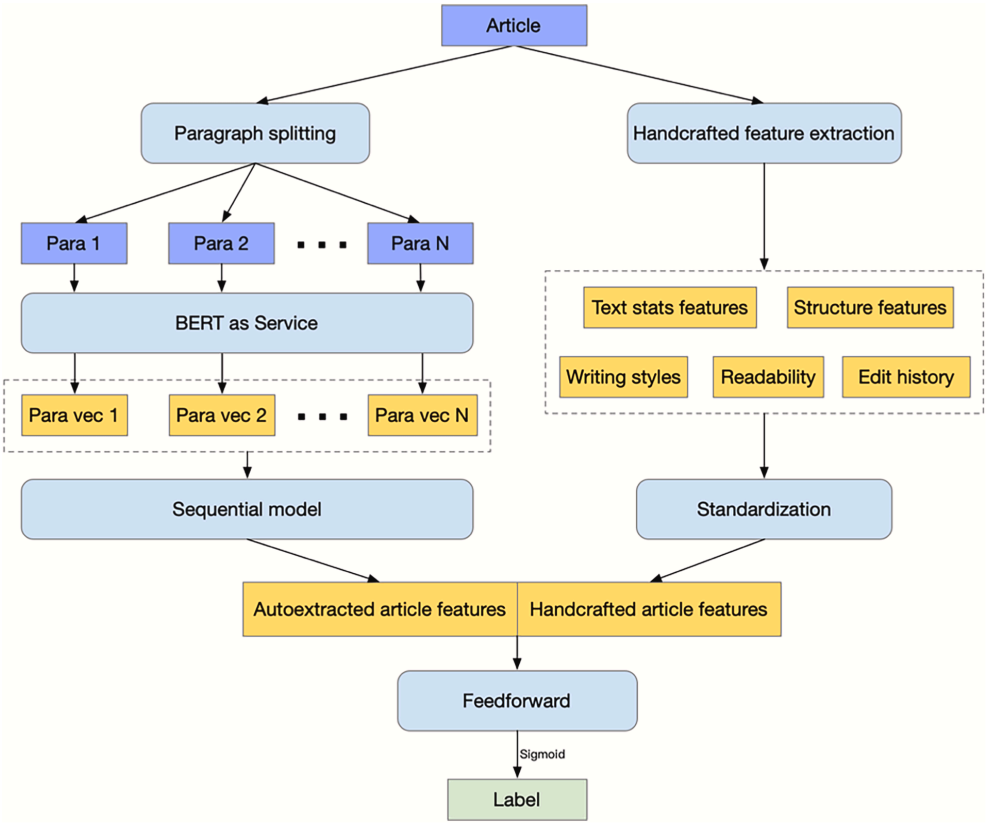


Figure 2.18: Quality flaw classification model based on fused features. (Wang et al., 2021)

2.4 Summary

In this chapter, we explore the flow of research conducted from the perspective of NLP on news quality classification tasks. This thesis will contribute to this flow by combining manual stylistic and automatic contextual features and incorporating traditional machine learning models into deep neural networks.

In this chapter, we deliberately did not mention the important work of Choi et al. (2021) as we will describe it in the next chapter.

3 Methods

In the previous chapter, we examined what and how researchers have studied news quality prediction in natural language processing (NLP). In this chapter, we will specify the data and the models that we will use.

As mentioned in Chapter 1, the aim of this thesis is to develop a model to predict news articles' quality levels evaluated by readers automatically. For this purpose, we use the news corpus and the quality levels collected by Choi et al. (2021), which is the base of this thesis.

3.1 Data from Choi, Shin, and Kang (2021)

3.1.1 News Corpus

The news corpus that Choi et al. (2021) collected comprises a total of 1,500 Korean newspaper articles on 11 social issues. Table 3.1 shows the issues that the news articles address. The articles were published by 21 news brands and collected from Naver News[1] from August 2017 to August 2018.

3.1.2 Quality Levels

Choi et al. (2021) conducted an online survey, and a total of 7,810 respondents, controlled for gender and political ideology, rated the quality of articles they read on the 10-point scale, in which 10 means the highest quality. As a result, each of the articles got evaluated by more than 50 respondents. Next, the quality scores were transformed into z-scores and then averaged. Finally, the

[1] `https://news.naver.com`

Issue	**Articles**
Minimum wage policy	191
Comprehensive real estate holding tax	158
South–North Korean summit conference	183
Yemeni refugee problem on Jeju island	138
The president's constitutional amendment proposal	222
Fine-dust policy measures	71
Secret agreement on sexual slavery with the Japanese government	163
Resumption of the Shin-kori nuclear power plant construction	120
College Scholastic Ability Test reform	90
Repeal of the abortion law	58
Conscientious objection to military service	106
Total	1,500

Table 3.1: Number of news articles about each of the 11 issues

1,500 articles were grouped into five categories, from 1, very low, to 5, very high, by their averaged z-scores. Each of the five categories has 300 articles. This category, or quality level, is the target to predict.

3.1.3 Journalism Values

Another result of Choi et al.'s survey is the collection of seven journalism values of articles. In their survey, the respondents rated whether they strongly agree (7), agree (6), somewhat agree (5), neither agree nor disagree (4), somewhat disagree (3), disagree (2), or strongly disagree (1) with the statements provided:

- Factuality: The news article is based on facts.
- Readability: The news article is easy to read.
- Diversity: The news article addresses diverse perspectives.

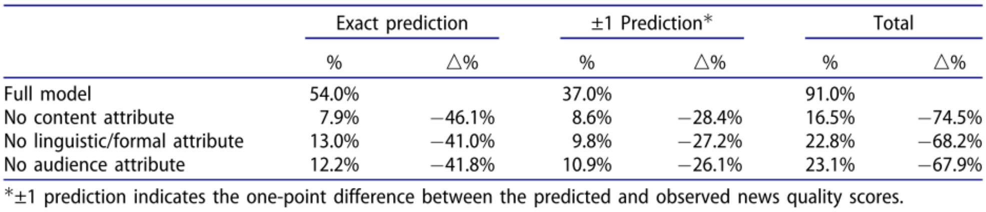
- Objectivity: The news article is objective.

- Sensationalism: The news article appeals to emotion.

- Depth: The news article is in-depth.

- Believability: The article is believable.

According to their experiment, six of these values (except sensationalism) are the strongest predictors of news quality. Their ablation study, presented in Table 3.2, showed that there would be a critical decrease (−74.5%) of accuracy if excluding journalism values (or content attributes). Despite this result, we take a different approach to that in Choi et al. (2021). In this thesis, we do not use journalism values and focus on linguistic features and their utilization.



	Exact prediction		±1 Prediction*		Total	
	%	△%	%	△%	%	△%
Full model	54.0%		37.0%		91.0%	
No content attribute	7.9%	−46.1%	8.6%	−28.4%	16.5%	−74.5%
No linguistic/formal attribute	13.0%	−41.0%	9.8%	−27.2%	22.8%	−68.2%
No audience attribute	12.2%	−41.8%	10.9%	−26.1%	23.1%	−67.9%

*±1 prediction indicates the one-point difference between the predicted and observed news quality scores.

Table 3.2: Accuracy change of test data by the elimination of individual news/audience attributes (Choi et al., 2021)

3.2 Linguistic Features

3.2.1 Justification of Using Linguistic Features Only

Although they determined the prediction model's performance, journalism values are difficult to obtain. As Choi et al. (2021) already pointed out, they require “the greater investment of respondents’ cognitive resources.” Moreover, a model’s applicability is also a problem. It is almost impossible to get the journalism values of many news articles published in real-time.[2] On the contrary, linguistic features can always be extracted from text.

In this thesis, we attempt to narrow the performance gap between “Full model” and “No content attribute model” in Table 3.2 by exploiting linguistic features of news article text and a broad range of techniques in natural language processing (NLP) and optimizing the model proposed by Choi et al..

3.2.2 Two Types of Linguistic Features

With an expression of “linguistic features,” we refer to two types of features: textual and contextual.[3,4]

3.2.2.1 Textual Features

Textual features capture the textual content of a news article, such as stylistic and sentiment features. Like “linguistic/formal attributes” in Choi et al.

[2] In addition to journalism values, quality scores also have limitations that can be obtained accurately without surveys. However, the quality level is predictable in our model, so we can extend the model using the highly reliable predictions as ‘silver dataset.’

[3] In NLP, the term *textual* is used in two ways: contrasting with *contextual* (Cignarella et al., 2020) and contrasting with *visual* (Goyal et al., 2021). The latter contains all the information obtained from the text. However, since our data is text unimodal, we use the former in a narrower sense.

[4] There are also two usages for the term *linguistic features*. Patil et al. (2020) and Imperial (2021) contrast “raw, contextualized, information-rich” features from BERT with “conventional, handcrafted linguistic features.” In this sense, contextual features are not linguistic. On the contrary, a series of studies on probing BERT (Conneau et al., 2018; Clark et al., 2019; Jawahar et al., 2019; Hewitt and Manning, 2019; Coenen et al., 2019; Alt et al., 2020) uses “linguistic features” to encode semantic and syntactic information in BERT. In this thesis, we adopt the latter usage in a broader sense.

(2021), they are obtained from hand-crafted rules. Chapter 4 will detail their subtypes.

3.2.2.2 Contextual Features

Contextual features capture the meaning of a word or a sentence. They are automatically extracted from a contextual embedding model such as BERT (Devlin et al., 2019). Chapter 5 will detail them.

3.3 Summary

In this chapter, we describe the data and the features we will use in the future. Then, in the following two chapters, we will build models to reflect linguistic features.

4 Ordinal Logistic Regression Models with Textual Features

This chapter investigates the roles of various textual features as factors affecting audiences' news quality evaluation.

4.1 Textual Features

To capture the writing style of articles and get independent variables, we preprocess news articles and morphologically analyze them using *KoNLPy*[1] (Park and Cho, 2014), a Python package for Korean natural language processing. We choose *MeCab*[2] as our tagger because it allows users to add customized morphemes to the tagging dictionary. POS tagging errors primarily committed by proper nouns such as names are corrected by adding them to the user dictionary. Furthermore, we extract textual features from the text concerning multiple sources, including Coh-Metrix, KOSAC Lexicon, and K-LIWC.

4.1.1 Coh-Metrix

Coh-Metrix (Graesser et al., 2004, 2011) provides measures of language and discourse for computational analyses of text characteristics. We obtain a total of 19 Coh-Metrix factors at the word level and the sentence level, and list them in Table 4.1.

[1] `https://konlpy.org/en/latest/`

[2] `https://bitbucket.org/eunjeon/mecab-ko-dic`

Level	Feature	Description or Examples
Word	Syl_per_wd	number of syllables per word
	Function_content.ratio	ratio of function words to content words
	Verbs	relative frequency of verbs
	Adjectives	rel. freq. of adjectives
	Adverbs	rel. freq. of adverbs
	Pronouns	rel. freq. of pronouns
	Pronouns_1P	rel. freq. of first-person pronouns
	Pronouns_3P	rel. freq. of third-person pronouns
	Connectives_Additive	더구나/MAJ, 또한/MAJ
	Connectives_Adversative	그러나/MAJ, 도리어/MAJ
	Connectives_Causal_Logical	고로/MAJ, 따라서/MAJ
	Connectives_Disjunctive	또는/MAJ, 혹은/MAJ
	Connectives_Identity	이른바/MAJ, 즉/MAJ
	Connectives_None	
	Connectives_Switch	그런데/MAJ, 어쨌든/MAJ
	Connectives_Temporal	으면서/EC, 자마자/EC
	Negations	안/MAG, 못/MAG
Sentence	Morph_per_sent	no. of morphemes per sentence
	Passive.constuctions	rel. freq. of 되/XSV

Table 4.1: Coh-Metrix features

4.1.2 KOSAC Lexicon

KOSAC (Jang et al., 2013; Kim et al., 2013), or Korean Sentiment Analysis Corpus, includes 7,713 sentence subjectivity tags and 17,615 opinionated expression tags manually annotated. Its lexicon[3] consists of morpheme *n*-grams classified by their sentiment polarity (POSITIVE, NEUTRAL, NEGATIVE, COMPLEX, None) and intensity (High, Medium, Low, None).

[3] http://word.snu.ac.kr/kosac

Feature	Description
Intensity_Medium	relative frequency of morphemes that likely appear when sentiment intensity is medium
nested_order_0	... when nested order is zero
nested_order_1	... when nested order is one
polarity_NEG	... when sentiment polarity is negative
polarity_None	... when sentiment polarity is none
polarity_POS	... when sentiment polarity is positive
subjectivity_polarity_POS	... when subjectivity polarity is positive
subjectivity_type_Argument	... when subjectivity type is argument
subjectivity_type_Judgment	... when subjectivity type is judgment

Table 4.2: KOSAC features

POS	K-LIWC	NE	Predicate	Others
NNP	posfeel	EV	obj_v	morph_main
VCP	hope	LC	exagg_v	morph_title
EP	anxiety	OG	unconfirm_v	INDR_QUOTE
SF	posfeel	PL	doubt_v	DR_QUOTE
	anger	PR	sub_v_assert	exclamation
	sad	PS	sub_v_pls	chinese
	cognitive		sub_v_exagg	english
	cause		sub_v_expect	foreignlang
	think		sub_v_concern	imagetable
	expect		sub_v_doubt	cosine_sim_byissue
	limit		sub_v_argu	no_reporter
	specu		sub_v_critic	email
	confirm		sub_v_warn	photographer
			sub_v_eval	byline
			sub_v_explain	byline_expertise
			specul_v	number
			eval_v	ordinal
				anonymity

Table 4.3: K-LIWC and other textual features

4.1.3 K-LIWC

K-LIWC (Lee and Yoon, 2005) is a Korean version of LIWC (Linguistic Inquiry and Word Count) (Pennebaker et al., 2001) that categorizes words grammatically and psychologically. Like KOSAC, it contains various sentiment features, as shown in Table 4.3.

4.1.4 Others

We calculate the relative frequency of part-of-speech tags from Sejong Tag Set (Kang and Kim, 2004), named entities, and predicate types (Park, 2006). We also measure text length from each news article and cosine similarity score between articles.

4.2 Ordinal Logistic Regression

In this chapter, the dependent variable is each article's quality level from 1 to 5. Since this variable is categorical and ordinal, we use ordinal logistic regression (Brant, 1990) (or Cumulative Link Model), as Table 4.4 indicates.

Variable	**Example**	**Model**
Binary	Positive-Negative	(Binomial) Logistic Regression
Nominal	Coffee-Tea-Water	Multinomial Logistic Regression
Ordinal	High-Mid-Low	Ordinal Logistic Regression

Table 4.4: Types of generalized linear models with categorical responses

Before training the regression model, we need to establish two assumptions to guarantee the validity of this model. First, we diagnose the multicollinearity among independent variables using variance inflation factors (VIF). We report that all VIF values are smaller than 10, which assures that our model is free from the problem of multicollinearity (O'Brien, 2007). Second, we conduct the Brant test (Brant, 1990), and it indicates our model does not violate the proportional odds assumption with a high *p*-value (0.4) after excluding nine variables: polarity_NEG, subjectivity_polarity_POS, EV, specul_v, Connectives_Disjunctive, Connectives_None, Connectives_Switch, Function_content.ratio, and sad. Appendix C includes the full results of the two tests.

4.3 Results

4.3.1 Feature Selection

We train the full ordinal regression model with 79 predictors using the `ordinal` package (Christensen, 2019) in R version 4.0.5 (R Core Team, 2021) and select 47 significant textual variables using the stepwise AIC (Akaike Information Criterion) method. We conduct the analysis of variance (ANOVA) between the full model and the selected model. As shown in Table 4.5, the *p*-value ($p = 0.9861$) justifies using the selected model because it does not significantly lose the explanatory power of the full model.

Model	**No. of parameters**	**AIC**	**logLik**	**LR.stat**	**df**	$P(> \chi^2)$
Selected	47	4526.9	-2216.5			
Full	79	4573.9	-2207.9	17.005	32	**0.9861**

Table 4.5: ANOVA results between the full model and the selected model

4.3.2 Impacts on Quality Evaluation

In the selected model, a total of 15 out of 47 independent variables are statistically significant with a p-value < 0.01. As reported in Table 4.6, six variables have positive coefficients, and nine variables have negative coefficients. Since we converted all independent variables to the same scale by standardizing them to have zero mean and unit variance (e.g. $[1, 2, 3]$ to $[-1, 0, +1]$), we can interpret their coefficients as their impacts on audience-rated news quality level. In the next section, we will discuss these factors in detail.

Feature (Positive)	Coefficient	Feature (Negative)	Coefficient
cosine_sim_byissue	$+0.44739$	specu	-0.19973
imagetable	$+0.25556$	sub_v_exagg	-0.19086
number	$+0.24245$	PR	-0.18261
anger	$+0.16040$	Pronouns_3P	-0.17611
obj_v	$+0.15700$	Connectives_Adversative	-0.17392
VCP	$+0.14429$	hope	-0.16338
		photographer	-0.13969
		cause	-0.13499
		Connectives_Identity	-0.12334

Table 4.6: Statistically significant variables and their coefficients, ordered by magnitude

4.4 Discussion

In the previous section, we have found which linguistic factors play a crucial role in predicting audience-rated news quality levels. To investigate their effects more clearly, we visualize their distribution at each quality level using box-whisker plots in Figures 4.1 and 4.2. Each panel corresponds to one independent variable.

From this visualization, we can recognize that some variables have a more consistent effect than others. For example, in Figure 4.1, all five boxes in the obj_v and VCP panels are at the same height. That means their values are similar across quality levels; therefore, their impact is less distinct. On the other hand, the boxes in the other four panels go up as the quality level increases, which implies that their values correlate positively with quality levels. In this way, we can identify the most influential variables, four positive ones (cosine_sim_byissue, numbers, imagetable, and anger) and two negative ones (sub_v_exagg and hope). To analyze their effects, we classify these six factors into three groups:

1. cosine_sim_byissue,
2. numbers and imagetable,
3. anger, hope, and sub_v_exagg.

4.4.1 Effect of Cosine Similarity by Issue

As shown in Table 4.6, the cosine_sim_byissue factor has the greatest coefficient (+0.44379) among all independent variables. The cosine_sim_byissue factor is defined by the average cosine similarity between a given news article and the other articles of the same issue. To calculate cosine similarity values, we use TF-IDF weighted vector embeddings.

Each of the news articles on the same issue, say d, is converted into a

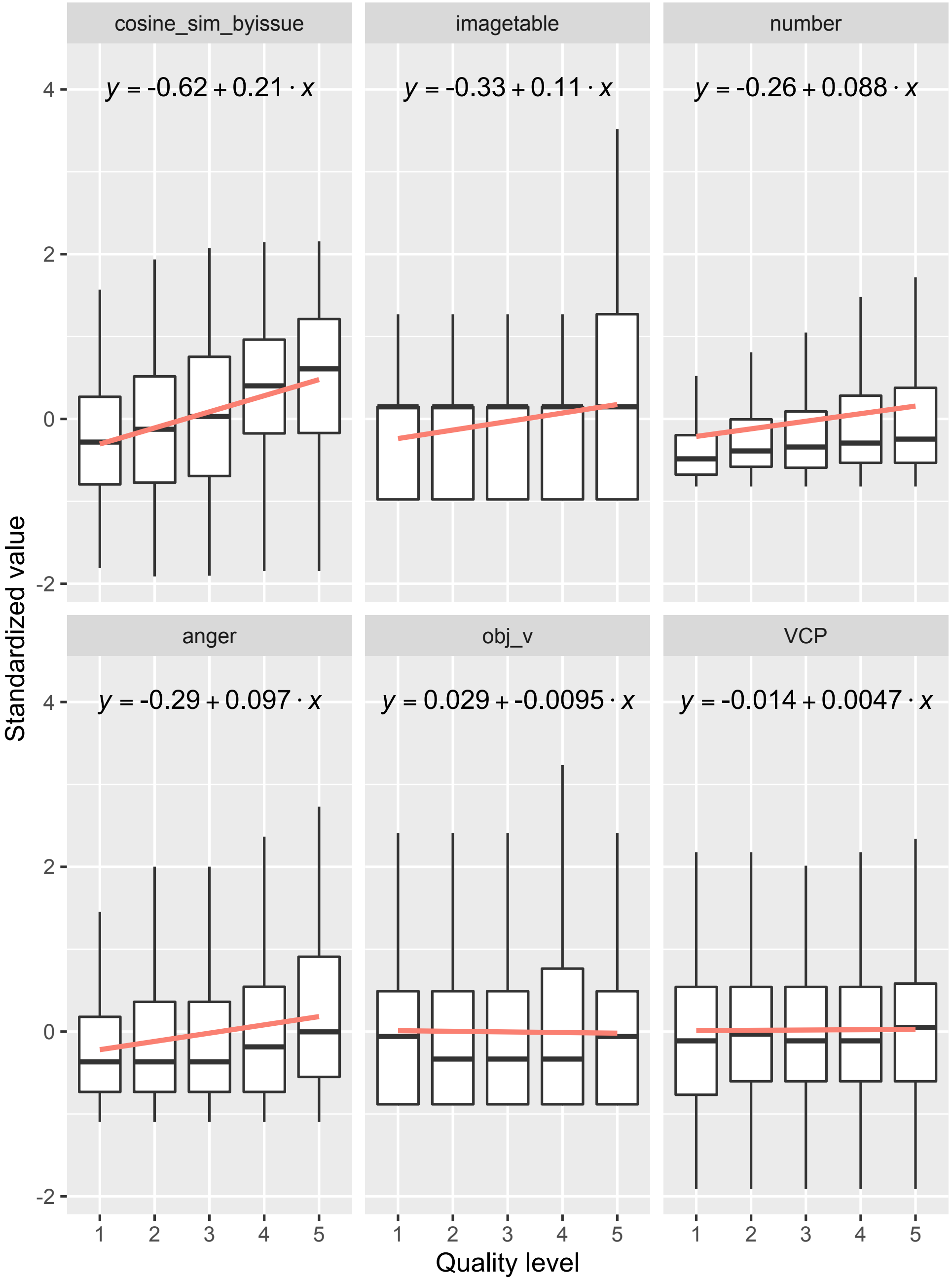


Figure 4.1: Distribution of each positive factor in news quality level

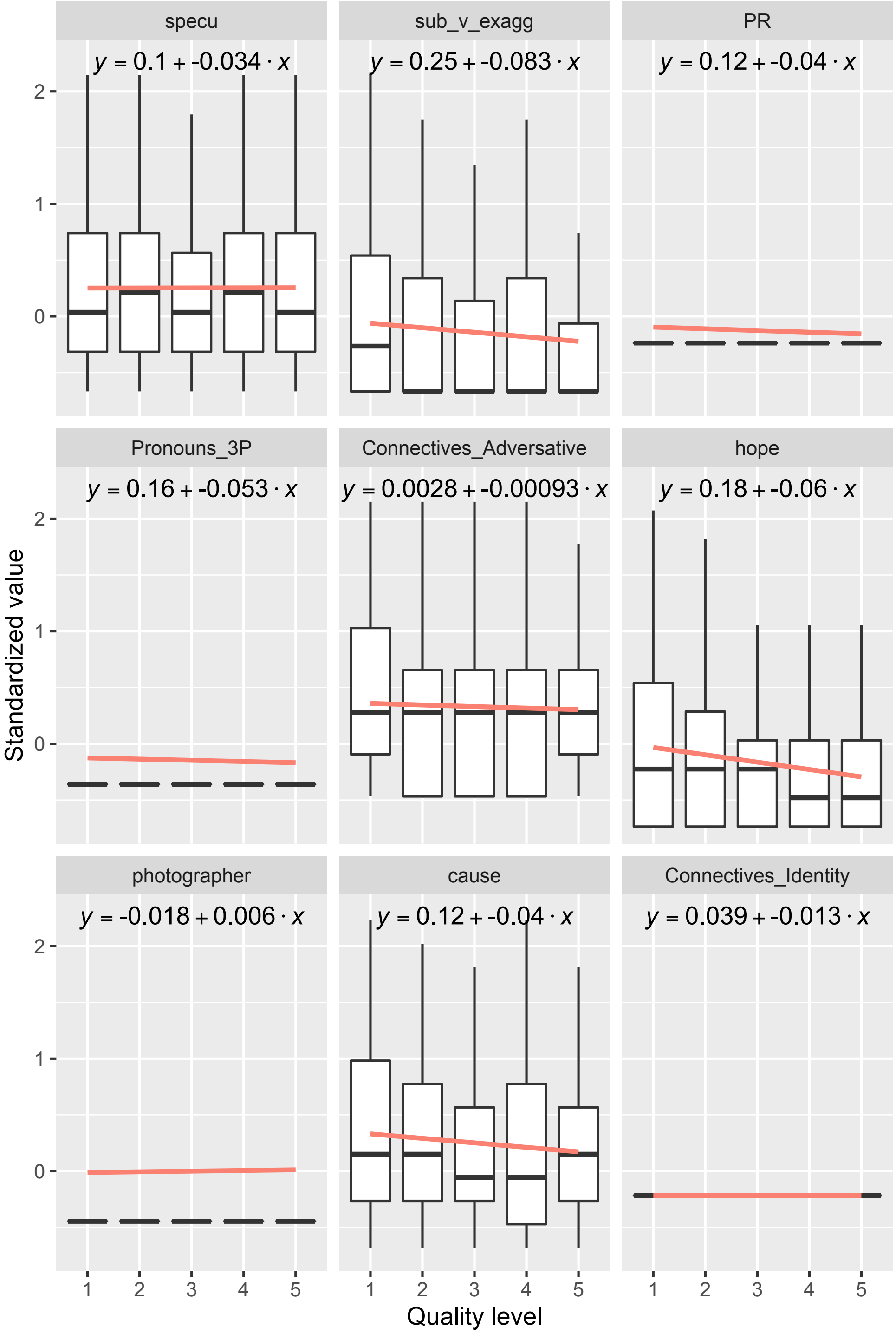


Figure 4.2: Distribution of each negative factor in news quality level

TF-IDF vector by Equation 4.1.

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \quad \text{for each } t \in \text{Vocabulary} \tag{4.1}$$

$$\text{TF}(t,d) = \log_{10}\left[1 + \text{count}(t,d)\right] \tag{4.2}$$

$$\text{IDF}(t) = \log_{10} \frac{[\text{the number of articles}]}{[\text{the number of articles containing } t]} \tag{4.3}$$

The term frequency $\text{TF}(t,d)$ measures how important a term t is in a document d, and the inverse document frequency $\text{IDF}(t)$, how much information the term t has in the corpus. Therefore, a high value of $\text{TF-IDF}(t,d)$ indicates the term t has much information and is important in the document t.

Now, given two vectors,

$$\vec{v} = (v_1, v_2, \cdots, v_d) \in \mathbb{R}^d \text{ and } \vec{w} = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^d,$$

we can calculate the cosine similarity between them. Let θ be the angle between $\vec{v}$ and $\vec{w}$. Then, from the fact that $\vec{v} \cdot \vec{w} = \sum_{i=1}^{d}(v_i w_i) = |\vec{v}||\vec{w}|\cos\theta$, the cosine similarity is defined as Equation 4.4.

$$\text{cossim}(\vec{v}, \vec{w}) = \cos(\theta) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{d}(v_i w_i)}{\sum_{i=1}^{d} v_i^2 \sum_{i=1}^{d} w_i^2} \tag{4.4}$$

Finally, we can get the cosine_sim_byissue factor. As we mentioned earlier in this subsubsection, it is the average cosine similarity between a given news article and the other articles of the same issue. Let there be n articles on the same issue, and $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ denote the TF-IDF article vectors. Then, the cosine_sim_byissue value of the j-th article is defined as Equation 4.5.

$$\mathsf{cosine\_sim\_byissue} = \frac{1}{n-1} \sum_{i=1, i \neq j}^{n} \text{cossim}(\vec{v}_i, \vec{v}_j) \tag{4.5}$$

Why do audiences consider a news article as of higher quality when it overlaps more with other articles, as Choi et al. (2021) already reported? Ideally, an article has the highest cosine_sim_byissue value if its vector is parallel to the sum (or the center of gravity) of all TF-IDF vectors on the same issue, which means that it is the concatenation of all the other articles.

The news articles in (3) and (4) on Shin-Kori nuclear power plant are good examples. The article in (3), whose cosine_sim_byissue values are 0.817, delivers stories from five sources—the industry, the academy, the ruling party, the opposite party, and the non-government organizations. On the contrary, the article in (4), providing only the ruling party's perspective, shows the lower value, 0.497. Notably, news audiences evaluated the former as very high and evaluated the latter as very poor.

(3) Article on Shin-kori plant with quality level 5 (very high)

[전략] 이번 결과에 여야 정치권은 물론 산업계 및 학계, 원전 지역민, 시민단체 등 이해당사자의 표정은 크게 엇갈렸다. 신고리 5.6호기 원전 건설업체 등 **산업계** 및 **원자력 학계**는 안도와 함께 환영을 뜻을 밝혔다. 이들은 ”원전에 대한 정확한 정보의 중요성을 확인했다”고 했다. 정치권에선 **더불어민주당**이 숙의민주주의를 통한 사회적 합의를 이뤄냈다는 점을 강조하며 대승적인 수용을 촉구한 반면, **야당**은 ”잘못된 탈원전정책을 철회하고 국론 분열을 유발한 데 대해 대국민 사과를 해야 한다”며 정부와 여당을 압박했다. **반원전 시민단체**는 ”아쉽지만 시민참여단의 판단을 존중한다”는 입장을 냈다.

(4) Article on Shin-kori plant with quality level 1 (very low)

[전략] **민주당 김경수 의원**은 지난달 13일 창원 강연에서 “중단하게 되면 2조6000억원의 예산이 날아가게 된다”며 “저는 제3의 대안도 있다고 생각한다”고 말했다. 그러면서 신고리 5 · 6호기는 계속 짓고 오래된 원전들을 닫으면 원전을 더 짓지 않는 것과 같은

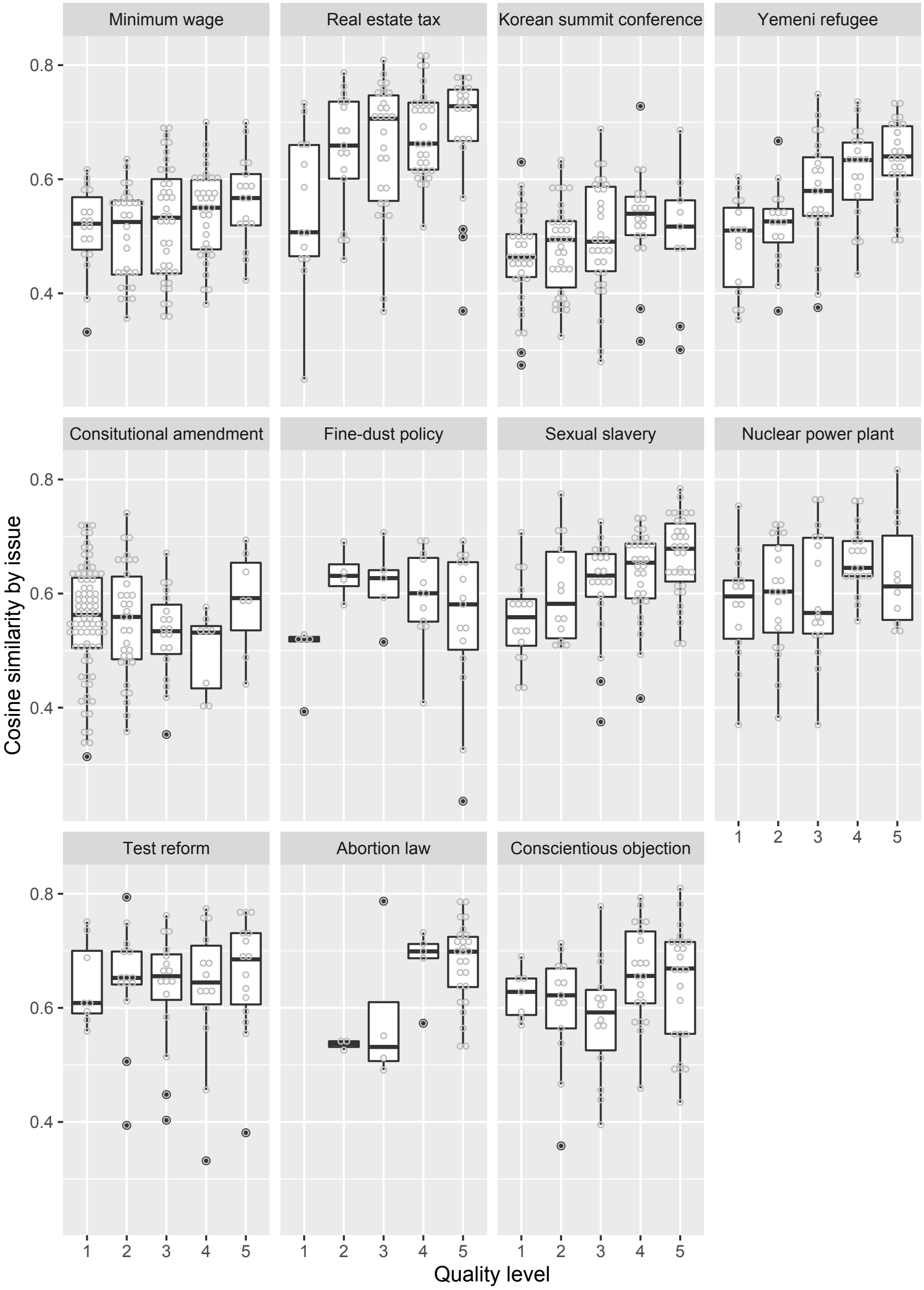


Figure 4.3: Distributions of cosine similarity

효과를 낼 수 있다는 것을 대안으로 소개했다. **정부 고위 관계자**는 “청와대와 정부는 중립성 시비 때문에 운신 폭이 거의 없었지만 더불어민주당은 정당으로서 충분히 역할을 할 수도 있었는데 공론화위 기간 중 신고리 5 · 6호기 문제는 물론 탈원전 논리를 개진하려는 의지를 별로 보이지 않았던 게 사실”이라고 말했다. **민주당 핵심 관계자**는 “잘못하면 여론조작 시비가 일 수도 있으니, 공론화 과정에 대해서 언급을 삼가자는 것이 지도부 스탠스였다”고 말했다.

4.4.2 Effect of Quantitative Evidence

The next largest coefficients in Table 4.6 are from the imagetable (+0.25556) and number (+0.24245) variables. These values are the relative frequency of images, tables, and numbers in a news article. Since their coefficients are positive, the more images and numbers an article has, the higher quality it gets.

We notice that numbers, images, and tables express quantitative evidence. Audiences consider a news article as objective if it has sufficient quantitative evidence. For example, the article in (5) demonstrates various precise numerical values and is evaluated as high quality (score 4).

(5) Article on real estate holding tax with quality level 4 (high)

[전략] A 씨가 서초래미안퍼스티지(2017년 공시가격 12억1600만 원), 마포래미안푸르지오(6억1700만 원), 용산한가람(6억9300만 원) 등 고가 3주택을 보유할 경우 종부세는 지난해 806만 원에서 내년에 1657만 원(105.48%)으로 오른다. A 씨는 종부세를 비롯해 재산세, 지방교육세, 농어촌특별세 등을 포함한 실질적인 세금 총액인 보유세 역시 같은 기간 1242만 원에서 2679만 원(115.63%)으로 증가한다. 이들 아파트는 올해 전년 대비 10.21 13.71%의 공시가격 상승률을 기록했고, 이를 2019년에도 반영해 세금을 추정했다.

종부세의 공정시장가액 비율은 현재 80%에서 내년에는 5%포인트 올라간다. 공정시장가액 비율은 세금을 부과하는 대상 금액을 정할 때 주택 공시가격을 얼마나 반영할지 정해 놓은 비율이다. 종부세의 경우 공시가격에서 6억 원(1주택자는 9억 원)을 뺀 금액에 이 비율(현재 80%)을 곱해서 구한다. 집값 합계가 10억 원인 주택 2채를 보유했다면 6억 원을 넘는 4억 원에 대해 80%, 즉 3억2000만 원에만 종부세가 부과된다. 고가 3 주택자 A 씨, 고가 2 주택자 B 씨(래미안퍼스티지, 래미안푸르지오), 1 주택자 C 씨(래미안퍼스티지)의 지난해 종부세를 비교하면 각각 806만원, 392만 원, 52만 원이다. 내년 공시가격 상승과 정부 종부세 개편안의 공정시장가액 비율 5%포인트 증가를 반영해 종부세를 추정하면 A 씨는 1657만 원(2017년 대비 105.48%), B 씨는 792만 원(102.04%), C 씨는 112만 원(115.38%)으로 나타난다. [후략]

4.4.3 Effect of Sentiment

In this subsection, we discuss the effect of sentiment expression on news quality. The related variables are sub_v_exagg (−0.19086), hope (−0.16338), and anger (+0.16040).

First, the sub_v_exagg variable and its negative coefficient mean that an article is more probably low-quality if it has exaggerations. For example, audiences evaluate the article in (8) containing many extreme nouns and verbs such as '광풍' *gust*, '한 몸을 던지겠다' *throw himself*, '구원투수' *relief pitcher*, and '혁명' *revolution* as very low quality (score 1). We explain this observation as exaggerative expressions reduce the credibility of a text.

(6) Article on constitutional amendment with quality level 1 (very low)

김문수 전 경기지사가 11일 자유한국당 6 · 13 지방선거 서울시장 후보 공천을 받고 출마를 공식 선언했다. 김 전 지사는 이날 서울 여의도 중앙당사에서 출마 기자회견을

열고 “대한민국을 좌파 **광풍**에서 구하고 자유민주주의 세력의 통합과 혁신을 위해 **한 몸을 던지겠다**”고 말했다. 김 전 지사는 “나라와 당이 큰 위기에 처했다. 문재인 정권의 좌향좌 · 정치보복을 **심판**해야 할 이번 선거에 (한국당이) 후보조차 제대로 내지 못할 처지가 됐다”고 말해 한국당의 구원투수 격으로 서울시장 후보에 나섰음을 강조했다. 김 전 지사는 출마 선언 내내 우파 정체성을 강조했다. 그는 “문재인 정권은 지금 **혁명**을 하고 있다”며 “국가가 민간기업의 주인 노릇을 하고 토지 사유권까지 침해하려 한다”고 비판했다. 김 전 지사는 공약으로 ‘수도이전 개헌’을 막겠다고 했다. 그는 “서울을 통일 수도, 동북아시아 자유의 수도, 세계 한민족의 수도로 발전시키겠다”고 했다. 정치권 관계자는 “우파 결집을 통한 핵심 지지층 결집이 시급하다는 전략을 세우고 **강경 발언을 쏟아낸** 것 같다”고 분석했다. 김 전 지사는 강한 우파 이미지로 인해 중도층 표심 잡기에 한계가 있다는 지적에 대해 “그런 우려를 느낀다”면서도 “인생에서 표를 많이 얻을 수 있을지를 생각하고 살지 않았다”고 말했다.

Second, more interestingly, our data shows that anger relates to high quality (+0.16040), and hope relates to low quality (−0.16338). We hypothesize that this result suggests that audiences sympathize with anger from victims and disapprove of uncritical hope. The article in (7) supports our assumption. It has the seventh-highest anger value among the 1,500 articles.[4] It explicitly requires a patient-centered approach to the military sexual slavery problem.

(7) Article on sexual slavery with quality level 5 (very high)

한국 정부가 ‘한-일 일본군 위안부 피해자 문제 합의’를 도출하는 과정에서 정작 ‘위안부’ 할머니들에게는 합의와 관련한 내용에 대해 구체적인 설명을 하지 않는 등 **‘피해자 중심적 접근’에 소홀**했던 것으로 드러났다. 한-일 일본군 위안부 피해자 문제 합의 검토 태스

[4] Although there are not many expressions of anger directly in this article, K-LIWC analyzed its anger value as very high. We hope to learn more about the internal mechanism of K-LIWC in the future.

크포스(TF · 이하 티에프)는 27일 검토 결과 보고서를 발표하고 “(외교부가) 최종적 · 불가역적 해결 확인, 국제사회 비난 · 비판 자제 등 한국 쪽이 취해야 할 조치가 있다는 것에 관해서는 (피해자들에게) 구체적으로 알려주지 않았다”고 밝혔다. 보고서를 보면 외교부는 한-일 국장급 협의 개시 결정 뒤 전국의 피해자 단체, 민간 전문가 등 2015년 한 해에만 모두 15차례 이상 피해자 및 관련 단체를 접촉했다. 티에프는 또 “외교부는 피해자 단체를 설득하는 게 중요하다는 인식을 가졌고, 협상을 진행하는 과정에서 피해자 쪽에 때때로 관련 내용을 설명했다”고 밝혔다. 하지만 ‘최종적 · 불가역적 해결 확인’ 등에 관해서는 제대로 알리지 않았고, 결과적으로 **피해자들의 이해와 동의를 이끌어내는 데 실패**했다고 티에프는 평가했다. 오태규 티에프 위원장은 이날 기자회견에서 “그냥 많이 접촉을 했다고 해서 그게 피해자 중심적 접근이라고 하면 안 되겠다. 피해자들의 목소리를 진짜 깊숙이 듣는 것이 필요하다”며 “그들이 어떤 요구를 하고 어떤 것을 바라는지 받아들이는 것이 중요하다”고 말했다. 2015년 12월28일 발표한 합의에 따라 한국 정부 주도로 발족한 ‘화해 · 치유재단’에 일본 정부가 10억엔(108억원)을 송금하기로 결정하는 과정에도 **피해자는 빠져 있었다**. 이날 티에프는 “일본 정부가 내는 돈이 10억엔으로 정해진 것은 객관적 산정 기준에 따른 것이 아니었다”며 “한 · 일 외교 당국의 협상과정에서 한국 정부가 피해자로부터 액수에 관해 의견을 수렴했다는 기록은 보지 못했다”고 밝혔다. 오 위원장은 “(액수 산정은) 무엇을 위해서, 어떤 용도로 얼마를 하는지 등 기준이 있어야 하는데, 그에 대해 논의했다는 어떠한 것도 확인하지 못했다”고 말했다.

The article in (8) is another example. We suppose that audiences perceive its hopeful expressions as excessive and then rate it very low.

(8) Article on South–North Korea with quality level 1 (very low)

북한과 지리적으로 맞닿은 강원도는 남북 교류사업을 적극적으로 추진하겠다고 밝혔다. 최문순 강원도지사는 29일 “남북정상회담이 성공적으로 개최된 것을 적극 환영한다”며 “다양한 남북 교류사업을 추진해 판문점선언을 뒷받침하겠다”고 밝혔다. 최 지사는

> 이날 오후 강원도청 신관 소회의실에서 기자회견을 열고 “강원도는 남북 정상이 합의한 판문점선언의 **선도적 실행의 장**이 될 수 있도록 모든 역량을 모아 남북 교류사업을 차질 없이 추진해 나가도록 하겠다”고 덧붙였다. 이에 따라 강원도는 관계기관과 협의해 동해 북부선 강릉∼제진 간 철도 연결 사업을 추진하고, 중·장기적으로 백마고지와 평강을 잇는 경원선 복원, 국도 31호선의 양구∼금강 구간 연결 사업 추진 등을 검토하기로 했다. 또 속초∼원산∼나진으로 운항하는 크루즈 항로를 열고, 설악(양양)∼원산(갈마)∼백두(삼지연) 등 남북 주요 관광지를 운항하는 항공 노선도 개설하는 방안을 추진키로 했다. 비무장지대(DMZ)를 **평화와 상생, 활력과 번영의 지역**으로 바꾸기 위한 사업도 추진된다. 철원 일대에 평화산업단지를 만들고, 정부의 한반도 신경제지도 추진계획과 연계해 설악산과 금강산을 국제관광자유지대로 조성해 **남북경제협력의 새로운 모델을 창출**할 계획이다. 강원도는 또 평창 동계올림픽 시설과 북한의 원산 마식령스키장 등을 활용해 2021년 동계 아시안게임을 남북이 공동으로 개최하는 방안을 추진하고, 말라리아 방역사업, 북한 강원도 지역의 결핵퇴치사업 등 기존에 추진하던 사업들도 확대 발전시키기로 했다.

4.5 Summary

In this chapter, we have analyzed the data statistically using textual features and ordinal regression, and discussed the factors influencing audience reception of news articles. We found that news audiences are more receptive to an article if it delivers more diverse perspectives, quantitative evidence, anger (mainly from victims), and less exaggeration or uncritical hope.

The limit of our study is that we did not deal with sufficiently more detailed facets of news quality, such as credibility, objectivity, and diversity. We also (not uncritically) hope that we get more insight on factors that seemed less consistent, such as obj_v and VCP.

5 Deep Transfer Learning Models with Contextual Features

In the previous chapter, we examined the influence of stylistic features on news quality. This chapter introduces another type of linguistic aspect of article texts, contextual features, then discusses the development of SBERT to quantify those features, and finally uses Deep Transfer Learning to predict news quality levels using contextual qualities. In addition, we compare the models in terms of their performance in news quality prediction.

Given a dataset, we normalize texts from news articles in the following way: First, HTML tags, special characters such as circled characters, parenthesized strings, and Chinese characters are removed. It is known that this process can improve a model's performance, especially for Korean.[1] Second, we tokenize the texts into sentences using *Korean Sentence Splitter*.[2] From these sentences, we extract contextual features.

5.1 Contextual Features from SentenceBERT

The expression "contextual features" contrasts with existing methods, such as TF-IDF, Word2Vec, and GloVe, representing the meaning of words as vectors, which are not contextual. Instead, they represent a single word as a single invariant vector to a word. Thus, a word always has the same vector in all possible contexts and cannot reflect polysemy. On the other hand, in

[1] https://github.com/monologg/KoELECTRA/blob/master/docs/preprocessing.md

[2] https://github.com/likejazz/korean-sentence-splitter

contextual language models, the vectors of the same word vary depending on the context. The context models include LSTM-based pre-trained language models such as ELMo (Peters et al., 2018) and Transformer-based models such as BERT (Devlin et al., 2019).

5.1.1 Necessity of Sentence Embeddings

BERT (Devlin et al., 2019) is the state-of-the-art model for NLP tasks. As we described in Subsubsection 2.3.1.2, its WordPiece tokenizer segments a sentence into subword tokens. Theoretically, the number of tokens can be arbitrary, but many pre-trained models limit the maximum length of a sentence to 512 or less for computational reasons.[3] This value is not much of a limit for general sentence classification tasks, but it invokes a problem for our data.

We count the word tokens in each article, and the result is as shown in the picture. As shown in Figure 5.1, nearly half of the articles consist of more than 512 tokens. Therefore, most existing pre-trained BERT models cannot read a complete article which will hinder the correct prediction of the quality of the article. This problem is severe for articles with higher quality is because articles with higher quality tend to be longer. On the other hand, Sentence-BERT, or SBERT, can solve these problems because it assigns a vector to a sentence rather than a word.

[3] In the multilingual BERT model, `max_seq_length` is set to 128.

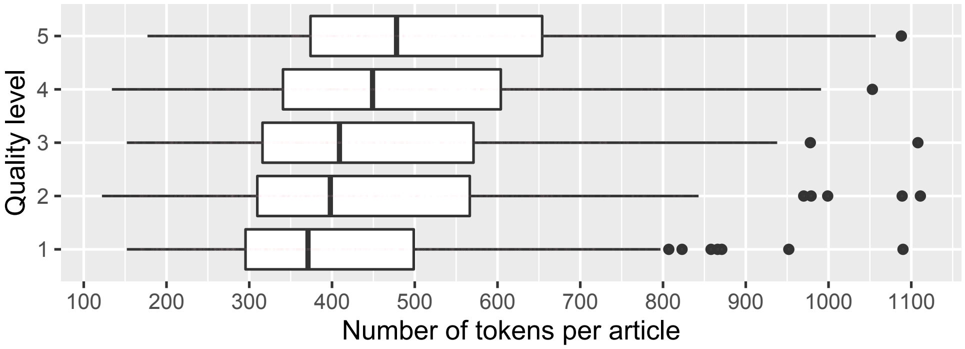


Figure 5.1: Distribution of news article lengths in tokens for each quality level

5.1.2 KR-SBERT

SBERT (Reimers and Gurevych, 2019) is a modification of the BERT network. It derives semantically meaningful sentences using siamese and triplet networks. It processes sentences typically with a maximum sequence length of 128 tokens, produces a fixed size sentence embedding of 768 dimensions by mean pooling, and updates the weights "such that the produced sentence embeddings are semantically meaningful and can be compared with cosine-similarity."

Since our dataset is written in Korean, we need a model that works well for the Korean language. For this purpose, we first prepare two pre-trained models, KR-BERT-MEDIUM (Lee et al., 2020) and KR-BERT-v40K,[4] and fine-tune them from KorNLI and KorSTS data (Ham et al., 2020).[5] We name these SBERT models KR-SBERT-MEDIUM-NLI-STS and KR-BERT-v40K-NLI-STS.

[4] `https://github.com/snunlp/KR-BERT`

[5] `https://github.com/kakaobrain/KorNLUDatasets`

Then, to examine the effects of the quantity and the quality of data, we augment our SBERT model on the KorSTS dataset using the In-domain approach suggested by Thakur et al. (2020) (see Figure 5.2). Finally, we fine-tune our KR-SBERT model on Klue-NLI data (Park et al., 2021), a refined version of KorNLI.

5.2 Deep Transfer Learning

This section shows how we use the contextual features obtained from the KR-SBERT models for news quality prediction. Given an article, we tokenize it into sentences and feed the sentences to the KR-SBERT model to get contextual features. As a result, we obtain a sequence of sentence vectors and use it as an input. Instead of training a new classifier from scratch, we transfer the KR-SBERT models to our task.

Figure 5.3 illustrates our transfer learning from SBERT. We add a `[CLS]` token to the first of sentence sequence, input the KR-SBERT embeddings to bidirectional transformers, and get a quality prediction as an output. With this approach, we contextualize sentence vectors.

To implement transfer learning, we borrow and modify a BertForSequenceClassification class using the *transformers* library (Wolf et al., 2020) by HuggingFace[6] and replace its weights with KR-SBERT's.

[6] `https://huggingface.co/transformers/`

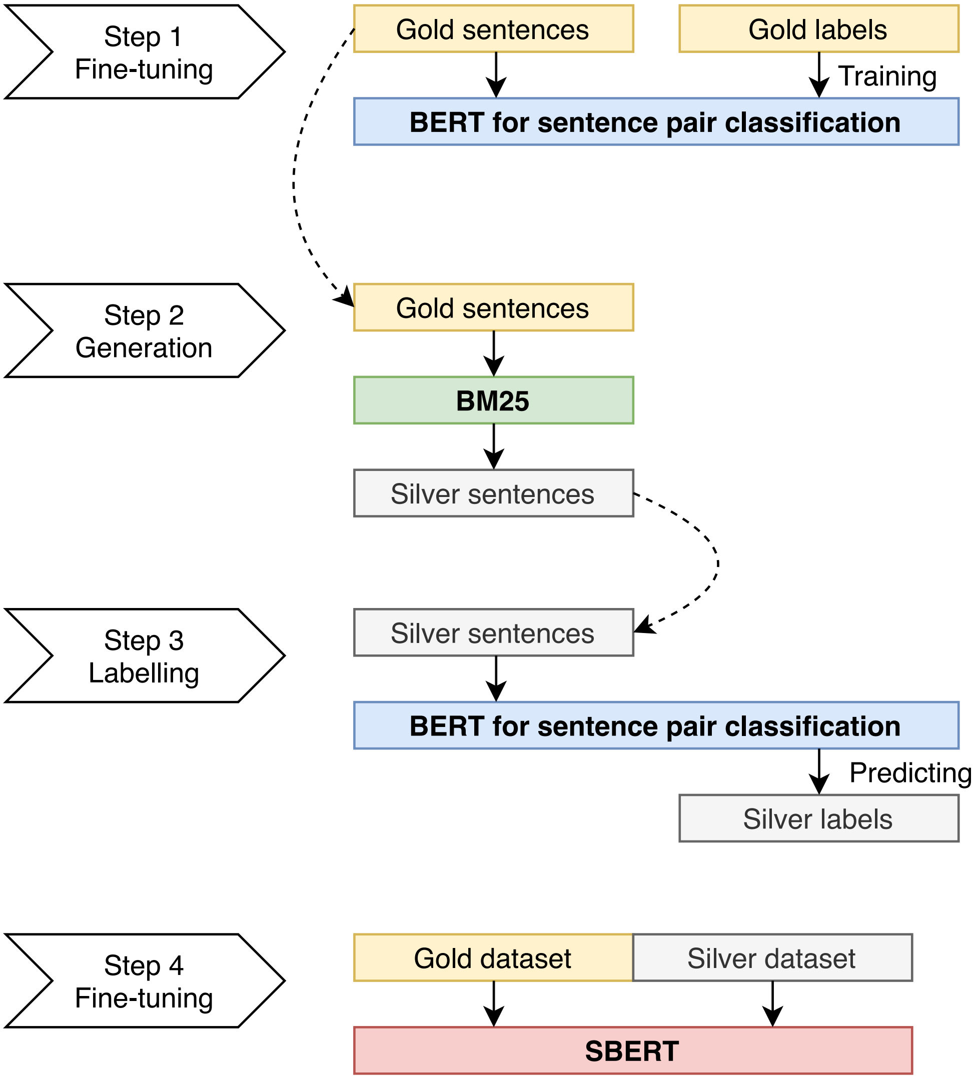


Figure 5.2: Augmented SBERT In-domain approach (Thakur et al., 2020)

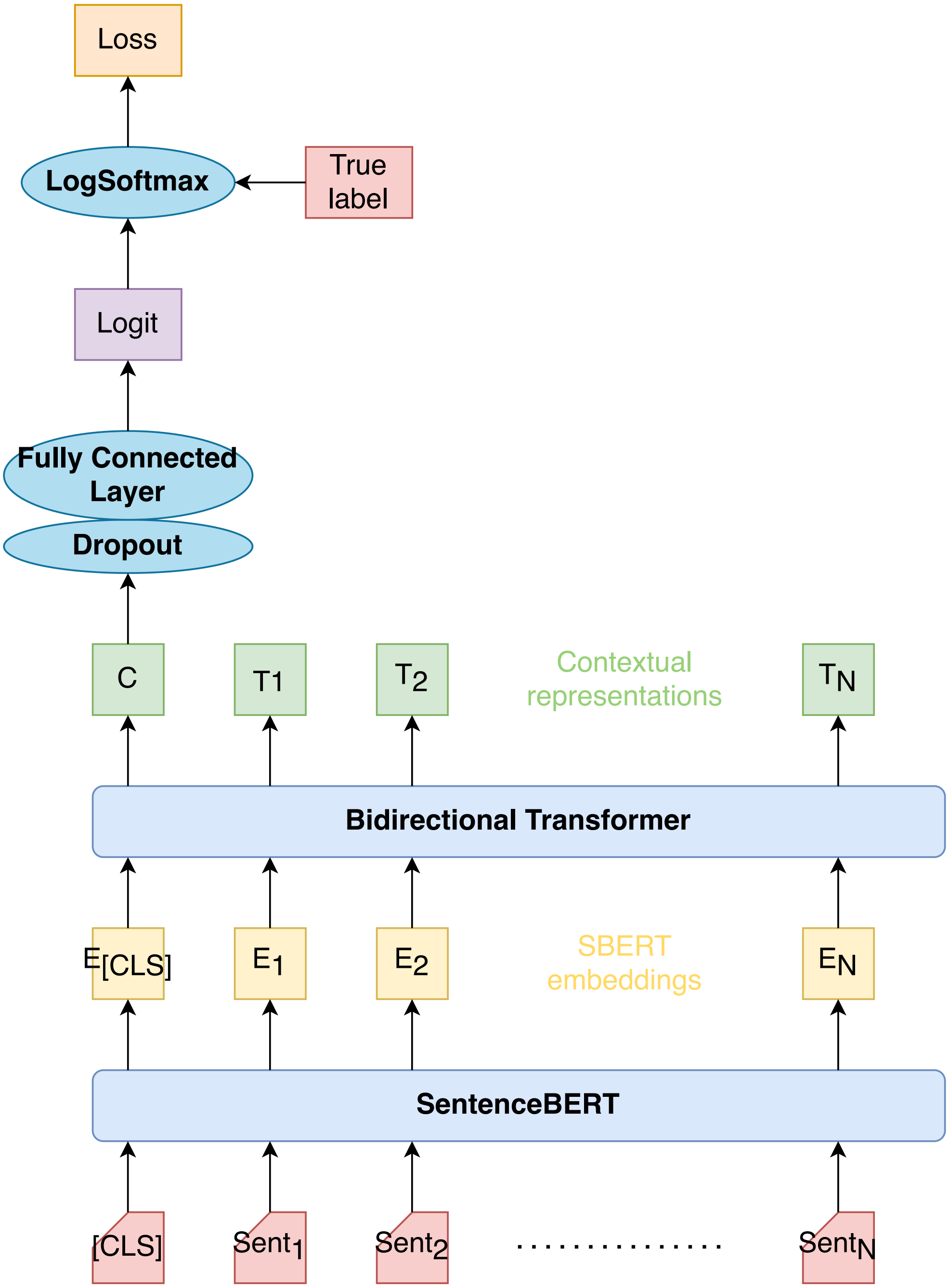


Figure 5.3: Architecture of deep transfer learning from SBERT embeddings

5.3 Results

5.3.1 Measures of Multiclass Classification

We measure the performance of our models in exact accuracy and one-off accuracy. Assume that we have a set of the predefined classes

$$K = \{1, 2, \cdots, k\},$$

a set of data,

$$D = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \cdots, (\vec{x}_n, y_n)\}$$

for each $\vec{x}_i \in \mathbb{R}^d$ and each $y_i \in K$ for $i = 1, 2, \cdots, n$, and a classification function mapping $\vec{x}_i \in \mathbb{R}^d$ to $f(\vec{x}_i) \in C$. Let $\hat{y}_i$ denote $f(\vec{x}_i)$ and then $(y_i - \hat{y}_i)$ is called a prediction residual.

Notation	Meaning
$\vec{x_i}$	i-th input data
y_i	i-th true label (class)
$\hat{y}_i$	i-th prediction
$y_i - \hat{y}_i$	i-th residual

Table 5.1: Data and Prediction

Let $\mathbb{1}$ be an indicator function and s be a statement. This means that $\mathbb{1}(s)$ is 1 if s is true, and as Equation 5.1

$$\mathbb{1}(s) = \begin{cases} 1 & \text{if } s \text{ is true} \\ 0 & \text{if } s \text{ is false} \end{cases} \tag{5.1}$$

Model	**Exact Accuracy**	**1-off Accuracy**
Random baseline	.2489	.6178
KR-BERT-MEDIUM	.3467	.7200
KR-BERT-v40K	.3333	.7067
KR-SBERT-MEDIUM-NLI-STS[7]	.3156	.6356
KR-SBERT-v40K-NLI-STS	.3967	**.7667**
KR-SBERT-v40K-NLI-augSTS	.4033	.7500
KR-SBERT-v40K-KlueNLI-augSTS	**.4233**	.7567

Table 5.2: Prediction performances of Transformer models

Then the exact accuracy and the one-off accuracy are defined as the following.

$$\begin{aligned}\text{Exact accuracy} &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(|y_i - \hat{y}_i| = 0\right) \qquad (5.2)\\ &= \frac{\text{(Number of \textit{correct} predictions)}}{\text{(Total number of predictions)}}\\ \text{One-off accuracy} &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(|y_i - \hat{y}_i| = 0 \ \lor \ |y_i - \hat{y}_i| = 1\right) \qquad (5.3)\\ &= \frac{\text{(Number of \textit{correct or adjacent} predictions)}}{\text{(Total number of predictions)}}\end{aligned}$$

5.3.2 Performances of news quality prediction models

Table 5.2 shows the performances of random baseline, KR-BERT models, and KR-SBERT models. The most crucial work, comparison with Choi et al. (2021), will be done in Chapter 6.

[7] Since KR-SBERT-MEDIUM has the same weights as KR-BERT-MEDIUM, we did not use the model in comparison.

- **Data and classification results**

  Let us assume that we have $n = 5$ data and $k = 3$ predefined classes, and our classifier f predicts classes for the data as the table below illustrates.

Input data	**True class**	**Predicted class**	**Absolute residual**	**Correct?**	**Adjacent?**
$\vec{x}_1$	$y_1 = 2$	$\hat{y}_1 = 2$	$\lvert y_1 - \hat{y}_1 \rvert = 0$	Yes	Yes
$\vec{x}_2$	$y_2 = 3$	$\hat{y}_2 = 1$	$\lvert y_2 - \hat{y}_2 \rvert = 2$	No	No
$\vec{x}_3$	$y_3 = 3$	$\hat{y}_3 = 2$	$\lvert y_3 - \hat{y}_3 \rvert = 1$	No	Yes
$\vec{x}_4$	$y_4 = 1$	$\hat{y}_4 = 1$	$\lvert y_4 - \hat{y}_4 \rvert = 0$	Yes	Yes
$\vec{x}_5$	$y_5 = 2$	$\hat{y}_5 = 3$	$\lvert y_5 - \hat{y}_5 \rvert = 1$	No	Yes

- **Calculating accuracies using prediction residuals**

$$\text{Exact Accuracy} = \frac{1}{5}(1+0+0+1+0) = \frac{2}{5} = 0.4$$

$$\text{One-off Accuracy} = \frac{1}{5}(1+0+1+1+1) = \frac{4}{5} = 0.8$$

- **Calculating accuracies using a confusion matrix**

  Orange indicates that the true label is equal to the predicted labee and yellow, the true label is adjacent to the predicted label.

		Predicted		
		1	**2**	**3**
True	**1**	1	0	0
	2	0	1	1
	3	1	1	0

Exact Accuracy

$$= \frac{1+1+0}{(1+0+0)+(0+1+1)+(1+1+0)}$$
$$= \frac{2}{5}$$
$$= 0.4$$

		Predicted		
		1	**2**	**3**
True	**1**	1	0	0
	2	0	1	1
	3	1	1	0

One-off Accuracy

$$= \frac{(1+1+0)+(0+1)+(0+1)}{(1+0+0)+(0+1+1)+(1+1+0)}$$
$$= \frac{4}{5}$$
$$= 0.8$$

Figure 5.4: Toy example of calculating exact and one-off accuracies

5.4 Discussion

From Table 5.2, we can see that SBERT models have the better performance than random baseline and BERT. Moreover, the difference among SBERT models are also observed. The differences in performance shown above suggest at least three things to us.

5.4.1 Effect of Data Size

First, we compare KR-SBERT-MEDIUM-NLI-STS with KR-SBERT-v40K-NLI-STS. The latter shows higher accuracy scores than the former. The difference between the two models comes from the size of KR-BERT. We observe that KR-BERT-v40K, which is larger than KR-BERT-MEDIUM, makes its SBERT version more effective. This is because the larger the model size, the larger the number of parameters, so the model can contain richer information.

5.4.2 Effect of Data Augmentation

The second thing we can look at in the experimental results is the effect of data augmentation. As shown in Table 5.2, the classification accuracy of fine-tuned models in augSTS instead of STS rose from .3967 to .4043. This observation also corresponds to the results of Thakur et al. (2020) where data augmentation was first introduced. Larger data allow for fine-tuning of siamese networks through more sentence-pair relationships. More information can be obtained from more data, which affects increasing quality.

5.4.3 Effect of Data Refinement

Thirdly, we can observe the effect of data refinement. As shown in Table 5.2, the classification accuracy of the fine-tuned model in KlueNLI instead of NLI increased from .4033 to .4233. If the data is refined, fine-tuning is possible for more correct sentences. We contribute to increasing the quality of the model by providing more correct sentences when fint-tuning SBERT. In particular, it is worth noting that we have obtained these results even though the number of data in klueNLI is significantly smaller than that of KorNLI. We conclude that quality, as well as quantity of data, is essential.

5.5 Summary

In this chapter, we have selected and trained SBERT models to obtain contextual features to be used for news quality predictions. SBERT, which expresses the meaning of sentences, can process words longer than BERT, making it suitable for article data consisting of dozens of sentences. We have prepared the SBERT model on various pre-trained BERT and fine-tuning data. The results confirm that quality prediction performance increases when the larger BERT model is applied to refined and augmented data in turn.

6 Fusion Models Combining Textual Features with Contextual Sentence Embeddings

In the previous chapter, we show that automatic prediction of news quality is possible using contextual features. In this chapter, we will examine whether contextual features perform better when combined with stylistic features, and compare these results with those of Choi et al. (2021).

6.1 Model Fusion

In this section, we present two ways to fuse a model using contextual features and a model using stylistic features.

6.1.1 Feature-level Fusion: Concatenation

First, the fine-tuned BERT model uses the `[CLS]` token's embedding for sequence classification. We call this embedding a contextual feature vector. We modify BERT Transfer Learning Model by concatenating a stylistic feature vector into a contextual feature vector. Then the concatenated vector is input into a feed-forward neural network for classification as in BERT's fine-tuned models.

6.1.2 Logit-level Fusion: Interpolation

We suggest another method other than vector concatenation to combine stylistic and contextual features. We keep the two models, one logistic regression and one BERT transfer learning, from the two types of features but fuse their

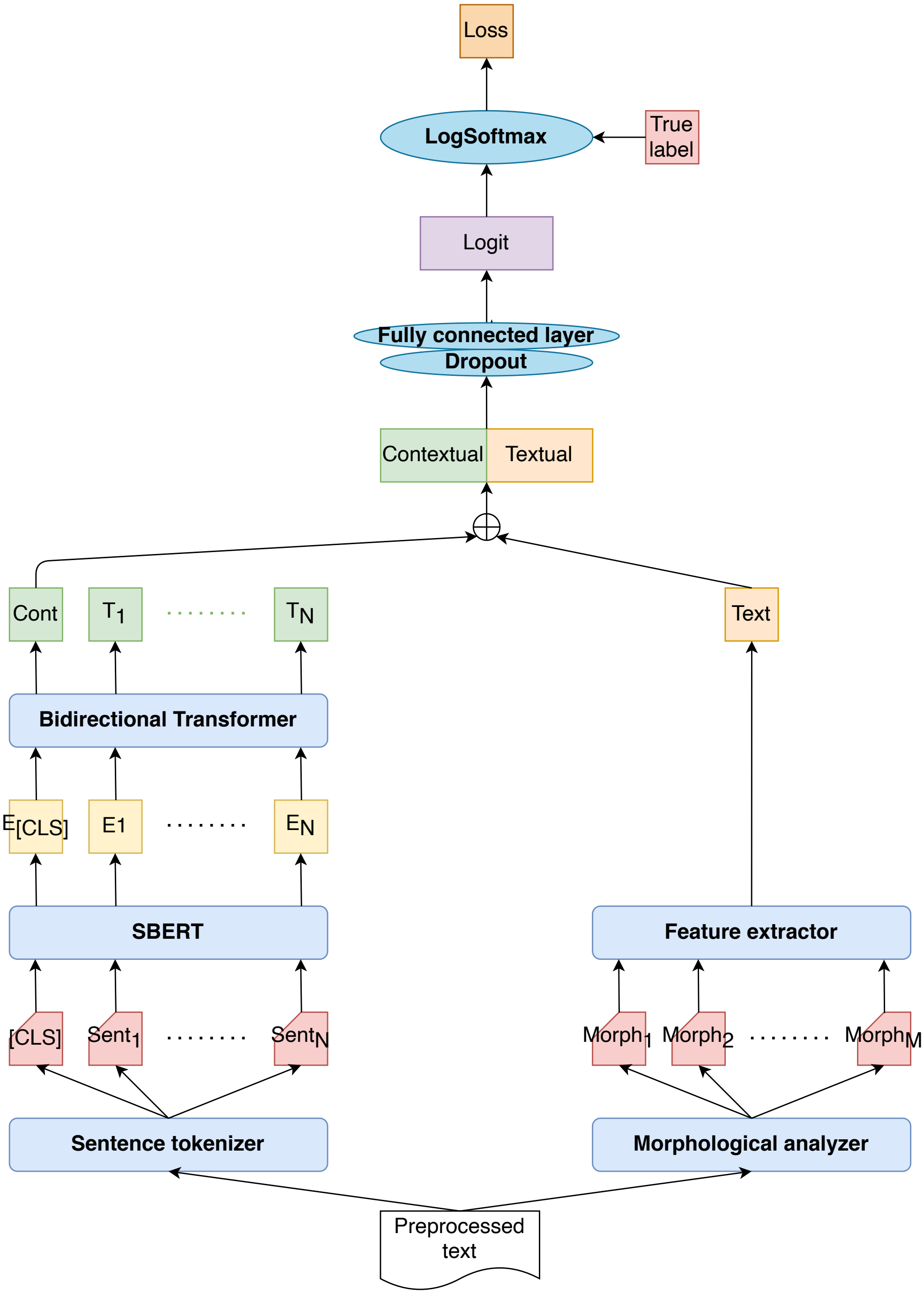


Figure 6.1: Architecture of feature-level model fusion

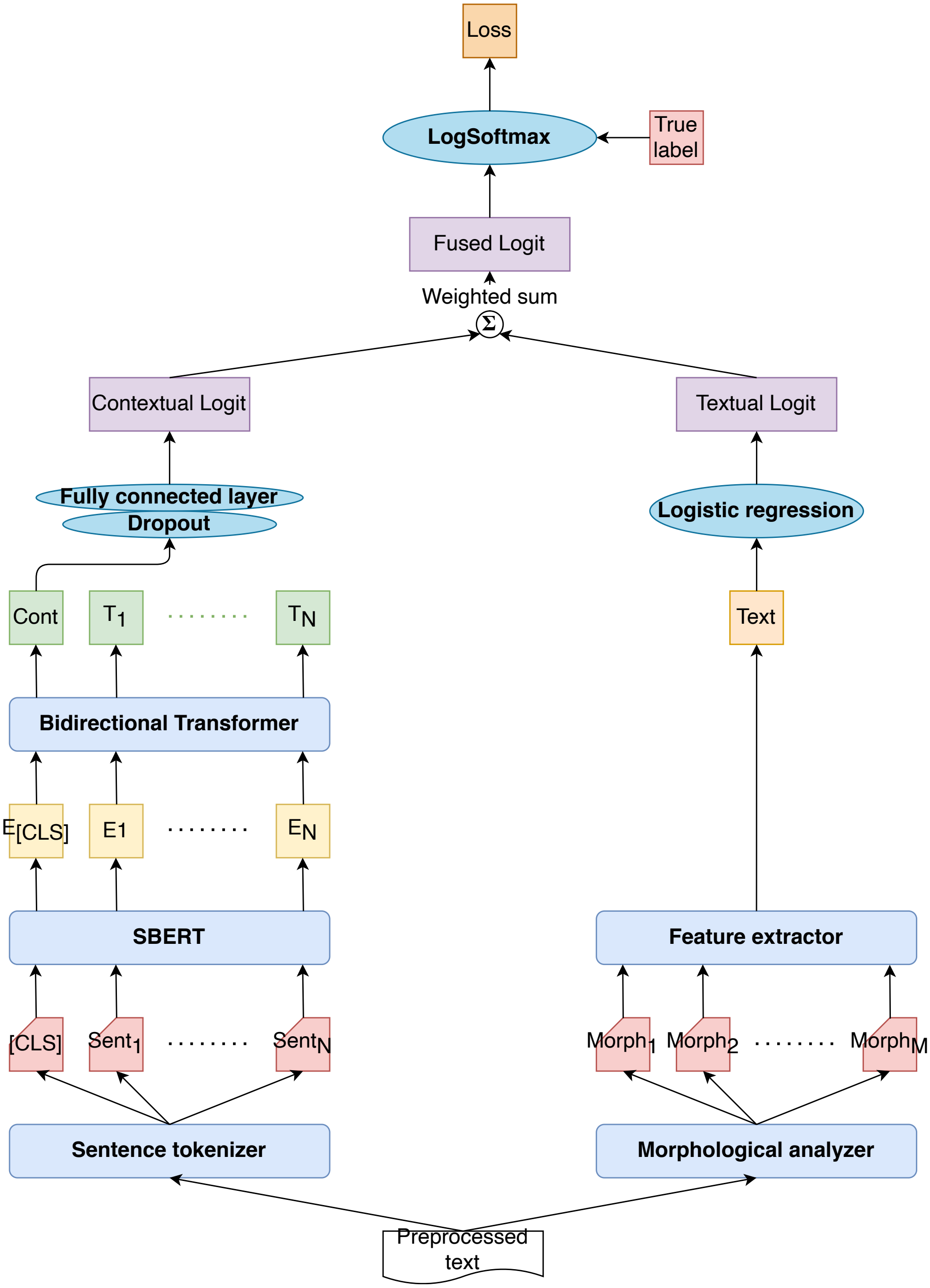


Figure 6.2: Architecture of logit-level model fusion

logit. That is, we use a weighted sum (6.1) of logits from two models.

$$\text{logit}_{\text{fusion}} = \alpha \times \text{logit}_{\text{contextual}} + (1 - \alpha) \times \text{logit}_{\text{stylistic}} \qquad (0 \leq \alpha \leq 1) \quad (6.1)$$

to find the proper value of α, we try different values from 0.1 to 1.0 for α. As a result, we find that value of 0.8 gives the best performances, as shown in Figure 6.3.

6.2 Results

After training two classifiers, we measure our results in exact accuracy, one-off accuracy, and macro average F_1.

6.2.1 Optimization of the Presentational Attribute Model

First, we optimize the Presentational Attribute Model of Choi et al. (2021) to compare its result to ours. For a fair comparison, we tune the hyperparameters of Feed-forward Neural Networks so that their presentational attributes can work effectively and list the results in Table 6.1.

6.2.2 Performances of News Quality Prediction Models

Table 6.2 summarizes the results of our experiments and Choi et al.'s. All models outperform the random baseline.

6.3 Discussion

In this section, we analyzed the experimental results in two main dimensions.

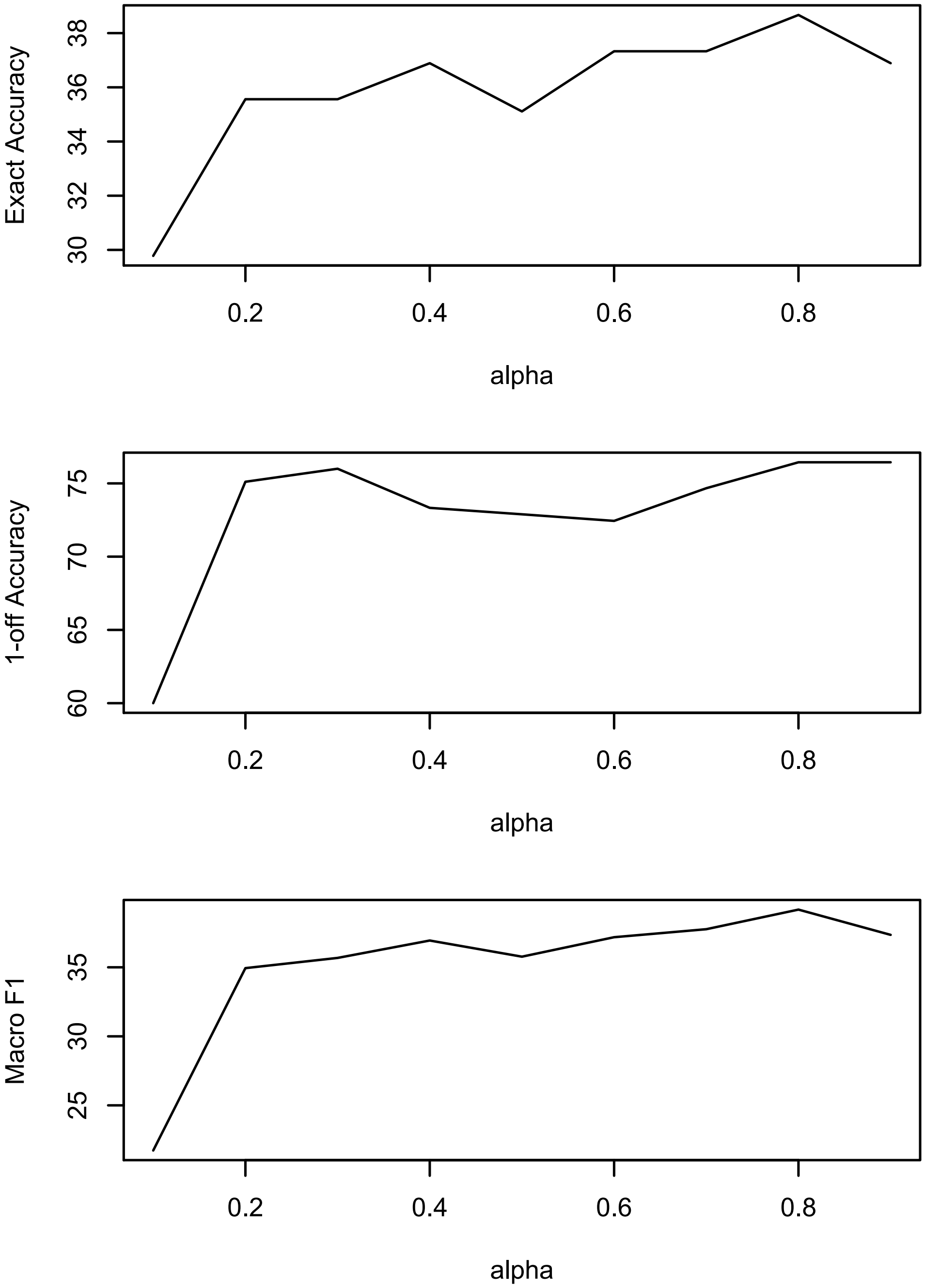


Figure 6.3: Performances of the logit-level fusion model for different values of α from 0.1 to 0.9.

No. of Hidden layers	**Hidden size**	**Dropout rate**	**Exact Acc.**	**1-off Acc.**
1	128	.50	.1666	.4633
1	128	.25	.2033	.5933
1	128	.00	.1900	.4900
1	64	.25	.2067	.5967
1	32	.25	.1600	.5567
2	64, 64	.25	.1633	.3300
2	64, 32	.25	**.2667**	**.6700**
2	64, 16	.25	.1967	.4633
3	64, 32, 16	.25	.1667	.5367

Table 6.1: Hyperparameter optimization of Presentational Attribute (Linguistic features) Model in Choi et al. (2021). We use AdaGrad as an optimizert and batch size of 128.

Model	**Exact Acc.**	**1-off Acc.**
This Thesis:		
Logistic Regression with stylilistic features	.3467	.7200
Transfer Learning with contextual features	.4033	.7500
Feature-level fusion	.4167	**.7867**
Logit-level fusion ($\alpha = 0.8$)	**.4200**	.7567
Choi et al. (2021):		
Full Model	<u>.5400</u>	<u>.9100</u>
No Content Attribute Model	.0790	.1650
Presentational Attribute Model (Our optimization)	.2667	.6700

Table 6.2: Prediction performances of fusion models

6.3.1 Effects of Fusion

First, stylistic features and contextual features work better together. Both Feature Concatenation and Logit Sum take effect. Feature Concatenation gets the highest one-off accuracy of .7867 among our models, and Logit Sum gets the highest Macro F_1 of .3918. This result suggests that we need both two linguistic types for news quality prediction. Even in the age of BERT, handcrafted stylistic features still prove their usefulness.

6.3.2 Comparison with Choi et al. (2021)

Second, our method improves prediction just using linguistic features only (one-off accuracy of .7867), comparing with Choi et al.'s Presentational Attribute Model (one-off accuracy of .6700). However, our model does not work as well as Full Model (one-off accuracy of .9100).

The results again show that journalistic values obtained through a survey are powerful in predicting news quality. However, we prove that linguistic factors can be utilized more effectively by increasing the accuracy of 67.00% to 78.67%.

Our results also suggest that the task of article quality classification can be applied to everyday life. Still, collecting journalistic values for new data all the time is time-consuming and costly. On the other hand, linguistic features are readily available to anyone with text. The accuracy of our model means that it is worth attempting to classify the quality of daily articles that cannot be surveyed.

6.4 Summary

In this chapter, we have presented a new method for the task of news quality prediction. We observe that contextual features extracted by a Sentence Transformer model are useful for quality classification, and handcrafted stylistic features also plays an essential role. We emphasize that contextual and stylistic factors should be considered together to select more relevant news.

7 Conclusion

This thesis posed the question: Can we predict the quality of news articles by only using the linguistic properties of the articles? To answer this question, first, we adopted the task of predicting news quality scores rated by audiences and the news and survey data used by Choi et al. (2021) in Chapter 3. Then we set the two types of linguistic features, namely stylistic features and contextual features. Next, we identified which stylistic features effectively explain the news quality scores in Chapter 4. To obtain adequate contextual features, we built a Transformer-based sentence representation model (KR-SBERT) and strengthened our model using data refinement and augmentation in Chapter 5. Finally, we incorporated two types of linguistic features using feature-level fusion and logit-level fusion in Chapter 6. When we evaluated our models, both fusion methods showed better prediction performance than non-fusion models and random baseline. Therefore the answer to the above question is "yes."

By developing a model that automatically predicts the quality of newspaper articles, our work can contribute to machine processing and providing high-quality articles, which is difficult for humans to process manually. In addition, the regression model selects features that play a significant role in classification performance, which can reveal linguistic factors that may affect the quality of articles and provide a new perspective on existing social science-oriented research. On the other hand, the KR-SBERT model developed in this work can also be applied to process long texts from other fields that have been difficult to process with BERT in natural language processing.

The limitation of our study is that our experimental results are not as promising as the performance of the full model in Choi et al. (2021), which includes not only stylistic features but also journalistic values as factors. With this as future work, we hope to examine more diverse linguistic factors, such as discourse features, and explore the potential for improvement in Transformer models.

References

Alhindi, T., Muresan, S., and Preotiuc-Pietro, D. (2020). Fact vs. Opinion: the Role of Argumentation Features in News Classification. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pages 6139–6149, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ali, R. and Ragb, H. K. (2019). Fused Deep Convolutional Neural Networks Based on Voting Approach for Efficient Object Classification. In *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, pages 335–339.

Alkoot, F. and Kittler, J. (2000). Multiple expert system design by combined feature selection and probability level fusion. In *Proceedings of the Third International Conference on Information Fusion*, volume 2, pages THC5/9–THC516 vol.2.

Alt, C., Gabryszak, A., and Hennig, L. (2020). Probing linguistic features of sentence-level representations in neural relation extraction. *arXiv*, pages 1534–1545.

An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., and Xie, X. (2019). Neural news recommendation with long- and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345, Florence, Italy. Association for Computational Linguistics.

Apte, C., Damerau, F., and Weiss, S. M. (1994). Towards language independent automated learning of text categorization models. In *SIGIR'94*, pages 23–30. Springer.

Arapakis, I., Peleja, F., Barla Cambazoglu, B., and Magalhaes, J. (2016). Linguistic Benchmarks of Online News Article Quality. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 4:1893–1902.

Aygunes, B., Cinbis, R. G., and Aksoy, S. (2021). Weakly supervised instance attention for multisource fine-grained object recognition with an application to tree species classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 176:262–274.

Bachmann, P., Eisenegger, M., and Ingenhoff, D. (2021). Defining and Measuring News Media Quality: Comparing the Content Perspective and the Audience Perspective. *International Journal of Press/Politics*, pages 1–29.

Biebricher, P., Fuhr, N., Lustig, G., Schwantner, M., and Knorz, G. (1988). The automatic indexing system air/phys-from research to applications. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 333–342.

Bogers, T. and van den Bosch, A. (2007). Comparing and evaluating information retrieval algorithms for news recommendation. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 141–144, New York, NY, USA. Association for Computing Machinery.

Borko, H. and Bernick, M. (1963). Automatic Document Classification. *Journal of the ACM (JACM)*, 10(2):151–162.

Bourgonje, P., Moreno Schneider, J., and Rehm, G. (2017). From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 84–89, Copenhagen, Denmark. Association for Computational Linguistics.

Brant, R. (1990). Assessing Proportionality in the Proportional Odds Model for Ordinal Logistic Regression. *Biometrics*, 46(4):1171–1178.

Cantador, I., Bellogín, A., and Castells, P. (2008). Ontology-based personalised and context-aware recommendations of news items. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 562–565.

Cao, Y., He, L., Ridley, R., and Dai, X. (2020). Integrating BERT and Score-based Feature Gates for Chinese Grammatical Error Diagnosis. *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 49–56.

Chiang, J.-H. and Chen, Y.-C. (2004). An intelligent news recommender agent for filtering and categorizing large volumes of text corpus. *International Journal of Intelligent Systems*, 19(3):201–216.

Choi, S., Shin, H., and Kang, S.-S. (2021). Predicting Audience-Rated News

Quality: Using Survey, Text Mining, and Neural Network Methods. *Digital Journalism*, 9(1):84–105.

Christensen, R. H. B. (2019). ordinal—regression models for ordinal data. R package version 2019.12-10. `https://CRAN.R-project.org/package=ordinal`.

Cignarella, A. T., Lai, M., Bosco, C., Patti, V., and Rosso, P. (2020). SardiStance @ EVALITA2020: Overview of the task on stance detection in Italian tweets. *CEUR Workshop Proceedings*, 2765:1–10.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What Does BERT Look at? An Analysis of BERT's Attention. pages 276–286.

Coenen, A., Reif, E., Kim, A. Y. B., Pearce, A., Viégas, F., and Wattenberg, M. (2019). Visualizing and measuring the geometry of BERT. *Advances in Neural Information Processing Systems*, 32(NeurIPS).

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:2126–2136.

Costera Meijer, I. and Bijleveld, H. P. (2016). Valuable Journalism: Measuring News Quality from a User's Perspective. *Journalism Studies*, 17(7):827–839.

Da San Martino, G., Yu, S., Barrón-Cedeño, A., Petrov, R., and Nakov, P. (2019). Fine-Grained Analysis of Propaganda in News Article. (4):5635–5645.

Danesh, A., Moshiri, B., and Fatemi, O. (2007). Improve text classification accuracy based on classifier fusion methods. In *2007 10th International Conference on Information Fusion*, pages 1–6.

Dang, Q. V. and Ignat, C.-L. (2016). Quality Assessment of Wikipedia Articles: A Deep Learning Approach. *ACM SIGWEB Newsletter*, (Autumn):1–6.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ferschke, O. (2014). *The Quality of Content in Open Online Collaboration Platforms: Approaches to NLP-supported Information Quality Management in Wikipedia.* PhD Thesis, Technischen Universität Darmstadt.

Foltz, P. W. (1990). Using latent semantic indexing for information filtering. In *Proceedings of the ACM SIGOIS and IEEE CS TC-OA Conference on Office Information Systems*, COCS '90, pages 40–47, New York, NY, USA. Association for Computing Machinery.

Foltz, P. W., Kintsch, W., and Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse processes*, 25(2-3):285–307.

Goodman, M. (1990). Prism: A Case-Based Telex Classifier. In *Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence*, pages 25–38.

Goyal, N., Sachdeva, N., and Kumaraguru, P. (2021). Spy the lie: Fraudulent jobs detection in recruitment domain using knowledge graphs.

Graesser, A. C., McNamara, D. S., and Kulikowich, J. M. (2011). Coh-Metrix: Providing Multilevel Analyses of Text Characteristics. *Educational Researcher*, 40(5):223–234.

Graesser, A. C., McNamara, D. S., Louwerse, M. M., and Cai, Z. (2004). Coh-Metrix: Analysis of Text on Cohesion and Language. *Behavior Research Methods, Instruments, & Computers*, 36(2):193–202.

Guda, B. P. R., Seelaboyina, S. B., Sarkar, S., and Mukherjee, A. (2020). NwQM: A Neural Quality Assessment Framework for Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 8396–8406, Online. Association for Computational Linguistics.

Ham, J., Choe, Y. J., Park, K., Choi, I., and Soh, H. (2020). KorNLI and KorSTS: New Benchmark Datasets for Korean Natural Language Understanding. *arXiv preprint arXiv:2004.03289*.

Hamill, K. A. and Zamora, A. (1980). The use of titles for automatic document classification. *Journal of the American Society for Information Science*, 31(6):396–402.

Hayes, P. J., Knecht, L. E., and Cellio, M. J. (1988). A news story categorization system. page 9.

Hayes, P. J. and Weinstein, S. P. (1990). Construe-TIS: A System for Content-based Indexing of a Database of News Stories. *Second Annual Conference on Innovative Applications of Artificial Intelligence*, pages 49–66.

Heaps, H. (1973). A theory of relevance for automatic document classification. *Information and Control*, 22(3):268–278.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4129–4138.

Hu, L., Xu, S., Li, C., Yang, C., Shi, C., Duan, N., Xie, X., and Zhou, M. (2020). Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4255–4264, Online. Association for Computational Linguistics.

Imperial, J. M. (2021). Knowledge-Rich BERT Embeddings for Readability Assessment. (1921).

Jang, H., Kim, M., and Shin, H. (2013). KOSAC: A Full-Fledged Korean Sentiment Analysis Corpus. In *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation, PACLIC 27*, pages 366–373.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *57th Annual Meeting of the Association for Computational Linguistics, ACL 2019*.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kang, B.-m. and Kim, H. (2004). Sejong Korean Corpora in the Making. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*.

Kar, G. (1975). *A distance measure for automatic sequential document classification*. PhD thesis, The Ohio State University.

Karimi, H., Roy, P., Saba-Sadiya, S., and Tang, J. (2018). Multi-source multi-class fake news detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1546–1557, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kim, M., Shin, H., Jang, H., and Jo, Y.-M. (2013). KOSAC (Korean Sentiment Analysis Corpus). In *Proceedings of the Korean Information Science Society Conference*, pages 650–652.

Kumar, M. A. and Gopal, M. (2010). A comparison study on multiple binary-class svm methods for unilabel text categorization. *Pattern Recognition Letters*, 31(11):1437–1444.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Lee, C. H. and Yoon, A. (2005). The Development of Korean Linguistic Inquiry and Word Count and its Characteristics. *Korean Journal of Cognitive Science*, 16:93–121.

Lee, S. (2021). *The Construction of a Korean Pre-Trained Model and an Enhanced Application on Sentiment Analysis.* PhD Thesis, Seoul National University.

Lee, S., Jang, H., Baik, Y., Park, S., and Shin, H. (2020). KR-BERT: A Small-Scale Korean-Specific Language Model. *arXiv preprint arXiv:2008.03979.*

Lewis, D. D. (1992a). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 37–50, New York, NY, USA. Association for Computing Machinery.

Lewis, D. D. (1992b). Evaluation of phrasal and clustered representations on a text categorization task. *Proceedings of the Fifteenth Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval, pages 37–50.

Lewis, D. D. and Ringuette, M. (1994). A comparison of two learning algorithms for text categorization.

Louis, A. (2012). Automatic Metrics for Genre-specific Text Quality. *Proceedings of the NAACL HLT 2012 Student Research Workshop*, pages 54–59.

Louis, A. (2013). *Predicting Text Quality: Metrics for Content, Organization and Reader Interest.* Doctoral dissertation, University of Pensylvania.

Louis, A. and Nenkova, A. (2013). What Makes Writing Great? First Experiments on Article Quality Prediction in the Science Journalism Domain. *Transactions of the Association for Computational Linguistics*, 1:341–352.

Maddalena, E., Ceolin, D., and Mizzaro, S. (2018). Multidimensional News Quality: A Comparison of Crowdsourcing and Nichesourcing. In *Proceedings of 6th International Workshop on News Recommendation and Analytics (INRA 2018).*

Maron, M. E. (1961). Automatic Indexing: An Experimental Inquiry. *Journal of the ACM (JACM)*, 8(3):404–417.

Marton, Y., Wu, N., and Hellerstein, L. (2005). On compression-based text classification. *Lecture Notes in Computer Science*, 3408:300–314.

Masand, B., Linoff, G., and Waltz, D. (1992). Classifying news stories using memory based reasoning. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, pages 59–65.

Mesgar, M. and Strube, M. (2018). A Neural Local Coherence Model for Text Quality Assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 4328–4339.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pages 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119.

Molyneux, L. and Coddington, M. (2020). Aggregation, Clickbait and Their Effect on Perceptions of Journalistic Credibility and Quality. *Journalism Practice*, 14(4):429–446.

O'Brien, R. M. (2007). A Caution Regarding Rules of Thumb for Variance Inflation Factors. *Quality & Quantity*, 41(5):673–690.

Park, E. L. and Cho, S. (2014). KoNLPy: Korean Natural Language Processing in Python. In *Annual Conference on Human and Language Technology*, pages 133–136. Human and Language Technology.

Park, J. (2006). The Front-Page Lead Story Analysis for the Development of News Evaluation Index (뉴스 평가지수 개발을 위한 신문 1면 머리기사 분석). In 2020 Media Committee (2020 미디어위원회), editor, *News Media* *of Korea (한국의 뉴스미디어 2006)*, chapter 2, pages 147–220. Korea Press Foundation (한국언론재단), Seoul, Korea.

Park, S., Moon, J., Kim, S., Cho, W. I., Han, J., Park, J., Song, C., Kim, J., Song, Y., Oh, T., Lee, J., Oh, J., Lyu, S., Jeong, Y., Lee, I., Seo, S., Lee, D., Kim, H., Lee, M., Jang, S., Do, S., Kim, S., Lim, K., Lee, J., Park, K., Shin, J., Kim, S., Park, L., Oh, A., Ha, J.-W., and Cho, K. (2021). Klue: Korean language understanding evaluation.

Patil, R., Singh, S., and Agarwal, S. (2020). BPGC at SemEval-2020 Task 11: Propaganda Detection in News Articles with Multi-Granularity Knowledge Sharing and Linguistic Features Based Ensemble Learning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1722–1731.

Peng, F., Schuurmans, D., and Wang, S. (2004). Augmenting naive bayes classifiers with statistical language models. *Information Retrieval*, 7(3):317–345.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). *Linguistic Inquiry and Word Count: LIWC 2001*, volume 71. Lawrence Erlbaum Associates, Mahway.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technolo-*

gies, NAACL-HLT 2018, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., and Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

Rau, L. F. and Jacobs, P. S. (1991). Creating segmented databases from free text for text retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–346.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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 2019*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rennie, J. D. M. (2005a). 20 newsgroups data set.

Rennie, J. D. M. (2005b). The log-log term frequency distribution.

Rezvani, N., Beheshti, A., and Tabebordbar, A. (2020). Linking Textual and Contextual Features for Intelligent Cyberbullying Detection in Social Media. *ACM International Conference Proceeding Series*, pages 3–10.

Rubin, V., Conroy, N., Chen, Y., and Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pages 7–17, San Diego, California. Association for Computational Linguistics.

Samarinas, C. and Zafeiriou, S. (2019). Personalized High Quality News Recommendations Using Word Embeddings and Text Classification Models. Technical report, EasyChair Preprint No, 1254.

Schütze, H., Hull, D. A., and Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 229–237, New York, NY, USA. Association for Computing Machinery.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47.

Sun, A., Lim, E.-P., and Liu, Y. (2009). On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, 48(1):191–201. Information product markets.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2020). Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. *arXiv preprint arXiv:2010.08240.*

Thorne, J., Chen, M., Myrianthous, G., Pu, J., Wang, X., and Vlachos, A. (2017). Fake news stance detection using stacked ensemble of classifiers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 80–83, Copenhagen, Denmark. Association for Computational Linguistics.

Tong, R. M. and Appelbaum, L. A. (1994). Machine learning for knowledge-based document routing (a report on the trec-2 experiment).

Urban, J. and Schweiger, W. (2014). News Quality from the Recipients' Perspective: Investigating Recipients' Ability to Judge the Normative Quality of News. *Journalism Studies*, 15(6):821–840.

Vildjiounaite, E., Kyllönen, V., Vuorinen, O., Mäkelä, S.-M., Keränen, T., Niiranen, M., Knuutinen, J., and Peltola, J. (2009). Requirements and software framework for adaptive multimodal affect recognition. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–7.

Volkova, S., Shaffer, K., Jang, J. Y., and Hodas, N. (2017). Separating Facts from Fiction : Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter. pages 647–653.

Wang, J., Li, Q., and Chen, Y. P. (2010a). User comments for news recommendation in social media. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 881–882, New York, NY, USA. Association for Computing Machinery.

Wang, J., Li, Q., Chen, Y. P., and Lin, Z. (2010b). Recommendation in Internet Forums and Blogs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 257–265, Uppsala, Sweden. Association for Computational Linguistics.

Wang, P., Li, M., Li, X., Zhou, H., and Hou, J. (2021). A hybrid approach to classifying Wikipedia article quality flaws with feature fusion framework. *Expert Systems with Applications*, 181:115089.

White, L. J. et al. (1975). A Sequential Method for Automatic Document Classification.

White, L. J., Petrarca, A. E., Crawford, L. G., Brinkman, B. J., and Mittal, S. (1977). CIRC II Data Base Classification. Technical report, OHIO STATE UNIV COLUMBUS DEPT OF COMPUTER AND INFORMATION SCIENCE.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-Art

Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wu, C., Wu, F., Ge, S., Qi, T., Huang, Y., and Xie, X. (2019). Neural news recommendation with multi-head self-attention. 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 6389–6394, Hong Kong, China. Association for Computational Linguistics.

Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., and Zhou, M. (2020). MIND: A Large-scale Dataset for News Recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Yang, Y. (1994). Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *SIGIR'94*, pages 13–22. Springer.

Young, S. R. and Hayes, P. J. (1985). Automatic classification and summarization of banking telexes. In *Proceedings of the Second Conference on Artificial Intelligence Applications*, pages 402–408.

Zaller, J. (2003). A New Standard of News Quality: Burglar Alarms for the Monitorial Citizen. *Political Communication*, 20(2):109–130.

Zhang, J., Jin, R., Yang, Y., and Hauptmann, A. G. (2003). Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 888–895. AAAI Press.

A List of Words Used for Textual Feature Extraction

A.1 Coh-Metrix Features

- Connectives_Causal_Logical: 고로/MAJ, 그래서/MAJ, 그러면/MAJ, 그러므로/MAJ, 그럼/MAJ, 그래야/MAJ, 따라서/MAJ, 다면/EC, 라면/EC, 면/EC, 로/EC, 으니/EC, 으면/EC, 으므로/EC, 아서/EC, 어서/EC

- Connectives_Disjunctive: 내지/MAJ, 또는/MAJ, 아니면/MAJ, 혹은/MAJ, 거나/EC, 든지/EC

- Connectives_Additive: 그리고/MAJ, 더구나/MAJ, 더욱이/MAJ, 또한/MAJ, 및/MAJ, 으며/EC, 고/EC

- Connectives_Adversative: 그러나/MAJ, 그렇지만/MAJ, 다만/MAJ, 단/MAJ, 도리어/MAJ, 오히려/MAJ, 하지만/MAJ, 더라도/EC, 라도/EC, 아도/EC, 어도/EC, 으나/EC, 지만/EC, 면서/EC, 으면서/EC

- Connectives_Identity: 소위/MAJ, 이른바/MAJ, 즉/MAJ

- Connectives_Switch: 그래도/MAJ, 그런데/MAJ, 근데/MAJ, 어쨌든/MAJ, 한편/MAJ, 는데/EC, 되/EC, 은데/EC

- Connectives_Temporal: 으면서/EC, 자마자/EC, 며/EC, 면서/EC, 자/EC, 으며/EC

- Negation: 안/MAG, 못/MAG, 없/VA, 못/VX, 않/VX, 아/VCN

- Passive Construction: 되/XSV, 된/XSV+ETM, 될/XSV+ETM, 됨/XSV+ETN, 돼/XSV+EC, 됐/XSV+EP

A.2 Predicate Type Features

- obj_v: 말했다, 밝혔다, 덧붙였다, 반문했다, 반대했다, 부인했다, 설명했다, 언급했다, 발표했다, 답했다, 진술했다
- sub_v_argu: 주장했다, 주장을 내놨다, 강조했다, 요구했다, 주문했다, 주장을 펼쳤다, 주장을 고수했다
- sub_v_pls: 당부했다, 호소했다, 바람을 나타냈다, 하소연 했다, 아쉬워했다, 안타까워했다
- sub_v_concern: 우려했다, 우려를 표명했다
- sub_v_assert: 단언했다, 단정했다, 일축했다, 고수했다
- sub_v_warn: 촉구했다, 경고했다, 으름장을 놨다, 압박을 가했다, 선언했다
- sub_v_critic: 지적했다, 꼬집었다, 비판했다, 비난했다
- sub_v_explain: 해명했다, 반발했다, 항변했다, 불쾌감을 표시했다, 목소리를 높였다
- sub_v_doubt: 의혹을 제기했다, 의문을 제기했다, 의아해 했다
- sub_v_expect: 시사했다, 내다봤다, 내비쳤다, 기대했다, 전망했다, 기대를 나타냈다

- sub_v_eval: 평했다, 평가했다

- sub_v_exagg: 라고도 했다, 말할 정도다, 털어놓았다, 토로했다, 귀띔했다, 벌어진 입을 다물지 못했다, 말을 제대로 잇지 못했다, 입을 모았다, 자조적인 태도를 보였다, 흥분을 감추지 못했다, 망설이다 한마디 더 붙였다, 한숨을 쉬었다, 혀를 내둘렀다, 완화된 어조로 나왔다, 신중한 반응을 보였다, 비판했다, 날을 세웠다, 불만을 터뜨렸다, 위협했다, 강조했다, 비난했다, 강조했다, 반발했다, 꼬집었다, 촉구했다, 부인했다, 우려했다, 요구했다, 허탈해했다, 목소리를 높였다, 유감을 표시했다, 지적했다, 항의했다, 기대했다, 선언했다, 일축했다, 관측했다, 선포했다, 호소했다, 맞섰다, 으름장을 놨다, 우려감을 표시했다, 직격탄을 날렸다, 공세를 퍼부었다

- specul_v: 보인다, 보이기도 했다, 관측되고 있다, 관측도 나오고 있다, 관측도 있다, 전망이다, 전망된다, 전망도 있다, 전망이 나온다, 전망까지 나온다, 전망이 제기되고 있다, 라고 봤다, 내다봤다, 예고된다, 예상된다, 예정이다, 추산된다, 추정된다, 추정하고 있다, 우려가 제기되고 있다, 우려마저 제기되고 있다, 우려가 나온다, 우려된다, 우려가 크다, 우려를 낳게 한다, 우려되는 실정이다, 우려가 커지고 있다, 우려하고 있다, 가능성이 있음을 드러냈다, 가능성도 거론되고 있다, 논란이 사그라지지 않을 듯하다, 시사했다, 시사하는 것이다, 관심이 모아지고 있다, 주장이 나오고 있다

- eval_v: 지적을 받고 있다, 지적도 받은 바 있다, 지적되고 있다, 지적이 나오고 있다, 지적도 나오고 있다, 지적이 나온다, 지적이 많다, 지적이 제기되고 있다, 지적이다, 비판이 나오고 있다, 비판이 거세다, 비판의 목소리가 높다, 평가를 받는다, 평가도 나온다, 평가된다, 것이 일반적이다, 개연성이 높다, 짐작하게 한다, 셈이다, 한 셈이 됐다, 해야 할 판이다, 했을

법 하다, 를 느낄 수도 있다, 목소리가 높았다, 목소리도 높아지고 있다

- unconfirm_v: 전해졌다, 알려졌다, 했다고 한다, 알려지고 있다, 라고 전했다, 해석할 수 있다는 것이다, 쫓겨났다는 것이다

- doubt_v: 의혹을 사고 있다, 의혹이 일고 있다, 의혹도 있다, 의혹의 눈길이 쏠리고 있다, 의문도 제기된다

- exagg_v: 문의전화가 빗발치고 있다, 폭풍 전야의 분위기였다, 한결같은 지적이다, 쇄도하고 있다, 찬사가, 쏟아지고 있다, 전국이 소용돌이치고 있다, 급전직하의 나락이다, 격찬이 쏟아졌다

B Codes Used in Chapter 4

B.1 Python Code for Textual Feature Extraction

```
import re
from numpy import mean
import pandas as pd

content = ('NNG', 'NNP', 'VA', 'VX', 'XR')
verbs = ('VV', 'VX', 'XSV')
adjectives = ('VA', 'XSA')
adverbs = ('MAG', )
pronouns = ('NP', )
negations = (
    ('안', 'MAG'),
    ('못', 'MAG'),
    ('없', 'VA'),
    ('못', 'VX'), # 못했
    ('않', 'VX'),
    ('아', 'VCN'), # 아니, 아닐
)
passives = (
    ('되', 'XSV'),
    ('된', 'XSV+ETM'),
    ('될', 'XSV+ETM'),
    ('됨', 'XSV+ETN'),
    ('돼', 'XSV+EC'),
    ('됐', 'XSV+EP'),
)
connectives = {
    'Causal_Logical': (
        r'\s(고로|그래서|그러면|그러므로|그럼|그래야|따라서)\_MAJ\s',
        r'\s(다면|라면면||므로|으니|으면|으므로)\_EC\s',
        r'\s([아어]서)\_EC\s(?!(?:[가-힣]+\_V는.*)|(?:\_JX))',
    ),
    'Disjunctive': (
        r'\s(내지|또는|아니면|혹은)\_MAJ\s',
        r'\s(거나|든지)\_EC\s',
    ),
    'Additive': (
        r'\s(그리고|더구나|더욱이|또한|및)\_MAJ\s',
        r'(?<=[있없]\_V[A-Z]{1})\s(으며)\_EC\s',
        r'(?<!"\_SSC)\s(고)\_EC\s(?![가-힣]+\_VX)',
    ),
    'Adversative': (
        r'\s(그러나|그렇지만|다만|단|도리어|오히려|하지만)\_MAJ\s',
        r'\s(더라도|라도|아도|어도|으나|지만)\_EC\s',
        r'\s(으?면서)\_EC\s도(?=\_JX)'
    ),
    'Identity': (
        r'\s(소위|이른바|즉)\_MAJ\s',
    ),
    'Switch': (
        r'\s(그래도|그런데|근데|어쨌든|한편)\_MAJ\s',
        r'\s(는데|되|은데)\_EC\s',
    ),
    'Temporal': (
        r'\s(으면서|자마자)\_EC\s',
        r'(?<!"\_SSC)\s(며)\_EC\s',
        r'(?<!"\_SSC)\s(면서)\_EC\s(?!도\_JX)',
        r'\s(자)\_EC\s(?!"\_SSC)',
        r'(?<![있없]\_V[A-Z]{1})\s(으며)\_EC\s',
    ),
    'None': (
        r'\s(어쩌다|역시|이르면)\_MAJ\s',
    ),
}

def isnegation(wd, tag):
    return any(wd.startswith(w) and tag.startswith(t) for w, t in negations)

def split_sentences(pos):
    sentences = []
    sent = []
    for wd, tag in pos:
        sent.append((wd, tag))
        if tag == 'SF':
            sentences.append(sent)
            sent = []
    return sentences

def is_in(pos, search_target):
    _pos = []
    pos_str = ['{}_{}'.format(wd, tag) for wd, tag in pos]
    for s in zip(['_']+pos_str[:-1], pos_str, pos_str[1:]+['_']):
        _pos.append(' '.join(s))

    match = lambda r, s: True if re.search(r, s) else False
    res = [any([match(r, s) for r in search_target]) for s in _pos]

    assert(len(pos) == len(res))
    return res

text_features = []
for issue_id, issue in data.items():
    for news_id, news in issue.items():
        pos = news['subtitle_pos'] + news['text_pos']
        sub_pos = news['subtitle_pos']
        body_pos = news['text_pos']
        text = news['기사부제목'] + ' ' + news['\uae30\uc0ac_\ubcf8\ubb38\ub0b4\uc6a9']
        news_features = {}
        news_features['issue_id'] = issue_id
        news_features['news_id'] = news_id

        news_features['Syl_per_wd']\
            = mean(list(map(len, re.sub(r'[^\uac00-\ud7a3 ]', ' ', text).split())))
        news_features['Nouns']\
            = mean([tag.startswith('N') for wd, tags in pos for tag in tags.split('+')])
        news_features['Verbs']\
            = mean([tag in verbs for wd, tags in pos for tag in tags.split('+')])
        news_features['Adjectives']\
            = mean([tag in adjectives for wd, tags in pos for tag in tags.split('+')])
        news_features['Adverbs']\
            = mean([tag in adverbs for wd, tags in pos for tag in tags.split('+')])
        news_features['Pronouns']\
            = mean([tag in pronouns for wd, tags in pos for tag in tags.split('+')])
        news_features['Pronouns_1P']\
            = mean([tag in ('NP') and wd in ('나', '내', '저희', '저')\
                for wd, tags in pos for tag in tags.split('+')])
        news_features['Pronouns_3P']\
            = mean([tag in ('NP') and wd in ('그', '그분', '그녀')\
                for wd, tags in pos for tag in tags.split('+')])
        news_features['Function-content ratio']\
            = sum(1 for wd, tags in pos for tag in tags.split('+') if tag not in content)
                / sum(1 for wd, tags in pos for tag in tags.split('+') if tag in content)
        news_features['Negations']\
            = mean([isnegation(wd, tag) for wd, tags in pos for tag in tags.split('+')])
        news_features['Morph_per_sent']\
            = mean(list(map(lambda x: sum(1 for wd, tags in x for tag in tags.split('+')),\
                [sub_pos] + split_sentences(body_pos))))
        news_features['Passive constuctions']\
            = mean([(wd, tag) in passives for wd, tags in pos for tag in tags.split('+')])
        for group, items in connectives.items():
            news_features['Connectives_'+group]\
                = mean(is_in(pos, items))
        text_features.append(news_features)
```

C Results of VIF test and Brant test

C.1 VIF Test in R

```
> car::vif(fit)
                morph_main                morph_title 
                  2.837040                   1.157623 
          intensity_Medium             nested_order_0 
                  3.027354                   1.672702 
            nested_order_1               polarity_NEG 
                  3.434747                   1.929556 
             polarity_None               polarity_POS 
                  2.323209                   1.251931 
 subjectivity_polarity_POS subjectivity_type_Argument 
                  5.138101                   4.351342 
subjectivity_type_Judgment                         EV 
                  1.576642                   1.064146 
                        LC                         OG 
                  1.373148                   1.492182 
                        PL                         PR 
                  1.111751                   1.182568 
                        PS                        NNP 
                  1.211257                   2.203353 
                       VCP                         EP 
                  1.560524                   2.941629 
                     obj_v                    exagg_v 
                  1.663865                   1.058919 
               unconfirm_V                    doubt_v 
                  1.213634                   1.064224 
              sub_v_assert                  sub_v_pls 
                  1.081239                   1.171609 
               sub_v_exagg               sub_v_expect 
                  5.323231                   1.148599 
             sub_v_concern                sub_v_doubt 
                  1.194833                   1.045112 
                sub_v_argu               sub_v_critic 
                  3.334815                   2.610492 
                sub_v_warn                 sub_v_eval 
                  1.327885                   1.153307 
             sub_v_explain                   specul_v 
```

```
               1.211018                   1.357412
                 eval_v                 INDR_QUOTE
               1.205930                   1.970833
               DR_QUOTE                 Adjectives
               1.598113                   1.703327
                Adverbs       Connectives_Additive
               1.482884                   1.841902
Connectives_Adversative Connectives_Causal_Logical
               2.200157                   1.313530
Connectives_Disjunctive       Connectives_Identity
               1.130394                   1.082921
       Connectives_None         Connectives_Switch
               1.098877                   1.172523
   Connectives_Temporal     Function_content.ratio
               1.508298                   3.532372
         Morph_per_sent                  Negations
               2.348938                   1.737998
   Passive.constuctions                   Pronouns
               1.211412                   1.773108
            Pronouns_1P                Pronouns_3P
               1.210126                   1.511483
             Syl_per_wd                      Verbs
               1.628319                   3.863148
            exclamation                    chinese
               1.166576                   1.105881
                english                foreignlang
               1.241672                   1.321674
             imagetable         cosine_sim_byissue
               1.562047                   2.138898
            no_reporter                      email
               1.176389                   1.179887
           photographer                     byline
               1.283665                   1.239598
       byline_expertise                    posfeel
               1.236574                   1.393100
                   hope                    anxiety
               1.242688                   1.189981
                  anger                        sad
               1.301139                   1.181155
              cognitive                      cause
```

```
                  5.293974                    1.386179
                     think                      expect
                  2.372513                    2.582965
                     limit                       specu
                  1.260765                    2.668279
                   confirm                      number
                  2.427890                    4.291127
                   ordinal                  anonymity
                  1.075554                    1.251848
```

C.2 Brant Test in R

```
> brant::brant(fit2)
------------------------------------------------------------
Test for                        X2      df      probability
------------------------------------------------------------
Omnibus                         229.5   225     0.4
morph_main                      2.86    3       0.41
morph_title                     5.1     3       0.16
intensity_Medium                0.94    3       0.82
nested_order_0                  0.93    3       0.82
nested_order_1                  0.77    3       0.86
polarity_None                   0.3     3       0.96
polarity_POS                    2.45    3       0.48
subjectivity_type_Argument      1.2     3       0.75
subjectivity_type_Judgment      3.2     3       0.36
LC                              1.06    3       0.79
OG                              2.66    3       0.45
PL                              2.75    3       0.43
PR                              2.94    3       0.4
PS                              2.11    3       0.55
NNP                             1.32    3       0.72
VCP                             0.71    3       0.87
EP                              1.19    3       0.75
obj_v                           2.16    3       0.54
exagg_v                         3.26    3       0.35
unconfirm_V                     6.23    3       0.1
doubt_v                         0       3       1
sub_v_assert                    2.18    3       0.54
sub_v_pls                       0.53    3       0.91
```

sub_v_exagg	1.17	3	0.76
sub_v_expect	5.99	3	0.11
sub_v_concern	1.4	3	0.7
sub_v_doubt	0	3	1
sub_v_argu	0.69	3	0.88
sub_v_critic	1.3	3	0.73
sub_v_warn	7.06	3	0.07
sub_v_eval	1.16	3	0.76
sub_v_explain	0.51	3	0.92
eval_v	3.67	3	0.3
INDR_QUOTE	1.71	3	0.64
DR_QUOTE	3.93	3	0.27
Adjectives	1.8	3	0.61
Adverbs	5.35	3	0.15
Connectives_Additive	3.65	3	0.3
Connectives_Adversative	1.76	3	0.62
Connectives_Causal_Logical	1.5	3	0.68
Connectives_Identity	2.9	3	0.41
Connectives_Temporal	0.7	3	0.87
Morph_per_sent	2.59	3	0.46
Negations	2.81	3	0.42
Passive.constuctions	6.59	3	0.09
Pronouns	1.66	3	0.65
Pronouns_1P	4.13	3	0.25
Pronouns_3P	1.09	3	0.78
Syl_per_wd	5.64	3	0.13
Verbs	1.31	3	0.73
exclamation	3.23	3	0.36
chinese	4.99	3	0.17
english	4.34	3	0.23
foreignlang	1.58	3	0.66
imagetable	3.19	3	0.36
cosine_sim_byissue	6.83	3	0.08
no_reporter	1.58	3	0.66
email	2.57	3	0.46
photographer	0.46	3	0.93
byline	3.15	3	0.37
byline_expertise	3.92	3	0.27
posfeel	1.23	3	0.75
hope	3.22	3	0.36

```
anxiety                         3.65    3       0.3
anger                           6.46    3       0.09
cognitive                       5.3     3       0.15
cause                           3.32    3       0.34
think                           4.65    3       0.2
expect                          0.28    3       0.96
limit                           1.78    3       0.62
specu                           6.4     3       0.09
confirm                         2.76    3       0.43
number                          1.81    3       0.61
ordinal                         1.09    3       0.78
anonymity                       4.37    3       0.22
------------------------------------------------------------

H0: Parallel Regression Assumption holds
```

D Codes Used in Chapter 6

D.1 Python Code for Feature-Level Fusion

```python
from transformers.models.bert.modeling_bert import *

class BertForSequenceClassificationConcat(BertForSequenceClassification):
    def __init__(self, config):
        super().__init__(config)
        self.num_labels = config.num_labels

        self.bert = BertModel(config)
        self.dropout = nn.Dropout(config.hidden_dropout_prob)
        self.classifier = nn.Linear(config.hidden_size + config.feat_size, config.num_labels)

        self.init_weights()

    # @add_start_docstrings_to_model_forward(BERT_INPUTS_DOCSTRING.format("batch_size, sequence
    # @add_code_sample_docstrings(
    #     tokenizer_class=_TOKENIZER_FOR_DOC,
    #     checkpoint="bert-base-uncased",
    #     output_type=SequenceClassifierOutput,
    #     config_class=_CONFIG_FOR_DOC,
    # )
    def forward(
        self,
        input_ids=None,
        attention_mask=None,
        token_type_ids=None,
        position_ids=None,
        head_mask=None,
        inputs_embeds=None,
        labels=None,
        output_attentions=None,
        output_hidden_states=None,
        return_dict=None,
        inputs_feats=None,
    ):
        r"""
        labels (:obj:`torch.LongTensor` of shape :obj:`(batch_size,)`, `optional`):
            Labels for computing the sequence classification/regression loss. Indices should be
            config.num_labels - 1]`. If :obj:`config.num_labels == 1` a regression loss is comp
            If :obj:`config.num_labels > 1` a classification loss is computed (Cross-Entropy).
        """
        return_dict = return_dict if return_dict is not None else self.config.use_return_dict

        outputs = self.bert(
            input_ids,
            attention_mask=attention_mask,
            token_type_ids=token_type_ids,
            position_ids=position_ids,
            head_mask=head_mask,
            inputs_embeds=inputs_embeds,
            output_attentions=output_attentions,
            output_hidden_states=output_hidden_states,
            return_dict=return_dict,
        )

        pooled_output = outputs[1]
        pooled_output = torch.cat((pooled_output, inputs_feats), axis=1)

        pooled_output = self.dropout(pooled_output)
        logits = self.classifier(pooled_output)

        loss = None
        if labels is not None:
            if self.num_labels == 1:
                #  We are doing regression
                loss_fct = MSELoss()
                loss = loss_fct(logits.view(-1), labels.view(-1))
            else:
                loss_fct = CrossEntropyLoss()
                loss = loss_fct(logits.view(-1, self.num_labels), labels.view(-1))

        if not return_dict:
            output = (logits,) + outputs[2:]
            return ((loss,) + output) if loss is not None else output

        return SequenceClassifierOutput(
            loss=loss,
            logits=logits,
            hidden_states=outputs.hidden_states,
            attentions=outputs.attentions,
        )
```

D.2 Python Code for Logit-Level Fusion

```
from transformers.models.bert.modeling_bert import *
from torch.nn import NLLLoss # for logistic regression

class BertForSequenceClassificationSum(BertForSequenceClassification):
    def __init__(self, config):
        super().__init__(config)
        self.num_labels = config.num_labels

        self.bert = BertModel(config)
        self.dropout = nn.Dropout(config.hidden_dropout_prob)
        # self.classifier = nn.Linear(config.hidden_size, config.num_labels)
        self.classifier0 = nn.Linear(config.hidden_size, config.num_labels) # BERT transfer
        self.classifier1 = nn.Linear(config.feat_size, config.num_labels) # Logistic regres
        if config.alpha:
            self.alpha = config.alpha
        else:
            self.alpha = .5

        self.init_weights()

    def forward(
        self,
        input_ids=None,
        attention_mask=None,
        token_type_ids=None,
        position_ids=None,
        head_mask=None,
        inputs_embeds=None,
        labels=None,
        output_attentions=None,
        output_hidden_states=None,
        return_dict=None,
        inputs_feats=None,
    ):
        r"""
        labels (:obj:`torch.LongTensor` of shape :obj:`(batch_size,)`, `optional`):
            Labels for computing the sequence classification/regression loss. Indices shoul
            config.num_labels - 1]`. If :obj:`config.num_labels == 1` a regression loss is
            If :obj:`config.num_labels > 1` a classification loss is computed (Cross-Entrop
        """
        return_dict = return_dict if return_dict is not None else self.config.use_return_di

        outputs = self.bert(
            input_ids,
            attention_mask=attention_mask,
            token_type_ids=token_type_ids,
            position_ids=position_ids,
            head_mask=head_mask,
            inputs_embeds=inputs_embeds,
            output_attentions=output_attentions,
            output_hidden_states=output_hidden_states,
            return_dict=return_dict,
        )

        pooled_output = outputs[1]

        pooled_output = self.dropout(pooled_output)
        logits0 = self.classifier0(pooled_output) # Bert Transfer learning
        logits1 = self.classifier1(inputs_feats) # logistic regression
        alpha = self.alpha
        logits = alpha * logits0 + (1-alpha) * logits1 # mean of logits

        loss = None
        if labels is not None:
            if self.num_labels == 1:
                #  We are doing regression
                loss_fct = MSELoss()
                loss = loss_fct(logits.view(-1), labels.view(-1))
            else:
                loss_fct = CrossEntropyLoss()
                loss = loss_fct(logits.view(-1, self.num_labels), labels.view(-1))

        if not return_dict:
            output = (logits,) + outputs[2:]
            return ((loss,) + output) if loss is not None else output

        return SequenceClassifierOutput(
            loss=loss,
            logits=logits,
            hidden_states=outputs.hidden_states,
            attentions=outputs.attentions,
        )
```

국문초록

뉴스 품질 예측을 위한 혼합 모형
— 텍스트 자질과 문장 임베딩 —

이 논문의 목표는 한국어 기사 품질을 예측하기 위한 언어 모형을 개발하는 것이다. 기사 품질 예측 과제는 최근 가짜뉴스 등의 범람으로 그 필요성이 대두되면서도 자연언어처리의 최신 기법이 아직 적용되지 못하는 실정에 있다. 이 논문에서는 이러한 한계를 극복하기 위해 문장의 의미를 표상하는 SBERT 모형을 개발하고, 기사의 언어학적 자질을 활용하여 품질 분류의 성능을 높일 수 있는지를 검토하고자 한다. 그 결과 기사의 가독성, 응집성 등의 텍스트 자질을 사용한 기계학습 모형과 SBERT에서 자동으로 추출된 문맥 자질을 사용한 전이학습 모형이 모두 선행연구의 심층학습 결과보다 높은 성능을 보였고, 구체적으로는 SBERT 학습시 훈련 데이터를 확장하고 정제할 때, 그리고 텍스트 자질과 문맥 자질을 함께 사용할 때 성능이 더욱 향상되는 것을 관측하였다. 이를 통해 기사의 품질에서 언어학적 자질이 중요한 역할을 하며 자연언어처리의 최신 기법인 SBERT가 언어학적 자질을 추출하고 활용하는 데 실질적으로 기여할 수 있다는 결론을 내릴 수 있다.

키워드: 컴퓨터언어학, 문장 임베딩, 뉴스 품질 예측, 혼합 모형, SBERT

학번: 2015-30035