Smart Techno (Smart Technology, Informatic, and Technopreneurship)
Volume 07, Number 02, 2025 pp. 34-44
E-ISSN : 2541-0679
Open Access: https://lppm.primakara.ac.id/jurnal/index.php/smart-techno/

SENTIMENT ANALYSIS OF COMMENTS ON INDONESIAN POLITICAL SPEECH VIDEOS ON YOUTUBE USING FASTTEXT

Bella Risma Khailla Savana[1*], Deni Arifianto[2], Lutfi Ali Muharom[3]
[1,2,3] Informatics, Universitas Muhammadiyah Jember, Jember, Indonesia

ARTICLE INFO
Article history:
Initial submission
03-07-2025
Received in revised form
15-10-2025
Accepted 18-10-2025
Available online 24-10-2025

Keywords:
Sentiment Analysis,
YouTube Comments,
FastText, Political Speech

DOI:
https://doi.org/10.59356/smart-techno.v7i2.159

ABSTRACT

The advancement of digital technology has transformed how society accesses and responds to political information, particularly through platforms like YouTube, which serve as arenas for public discourse. Comments on political speech videos often contain complex sentiments such as irony, slang, and code-mixing, which are difficult to identify using traditional sentiment analysis methods. This study aims to analyze public sentiment toward the Indonesian President's political speeches on YouTube from 2014 to 2024 using the FastText word embedding approach and to compare its performance with the TF-IDF + Logistic Regression method. The evaluation was conducted on three sentiment classes using automatically labeled data and oversampling experiments to address class imbalance. The results show that FastText achieved an accuracy of 76.82%, slightly higher than TF-IDF + Logistic Regression at 74.11%. Although the difference in accuracy is relatively small, the FastText model demonstrated more stable performance on informal texts and varied contexts. The use of oversampling helped balance predictions across classes without significantly improving accuracy. This study highlights the potential of FastText to enhance the effectiveness of Indonesian-language sentiment analysis, particularly for political comments on social media, while also revealing the limitations of automatic labeling that may affect classification outcomes.

1. INTRODUCTION

The development of information technology has transformed the way people access and respond to political information, particularly through platforms like YouTube, which have now become spaces for public discussion. Comments on political speech videos reflect various public opinions and sentiments toward the issues being addressed. However, these comments often contain complex expressions such as irony, slang, and code-mixing, which are difficult to detect using traditional sentiment analysis approaches.

Sentiment analysis is used to classify opinions expressed in text into positive, negative, or neutral categories, serving as an important tool for evaluating public responses to political speeches. One relevant approach is the use of word embedding techniques such as FastText, which can capture word meanings through subword-based vector representations. This technique is more adaptive in handling informal Indonesian text, including non-standard words and spelling variations.

Previous studies have examined sentiment analysis of political issues using machine learning and word embedding techniques such as Word2Vec, GloVe, and FastText. Putri et al. (2024) analyzed sentiment classification on the issue of Indonesia's capital relocation using CNN with GloVe and FastText-based representations. However, CNN-based methods tend to require very large datasets and are less effective in capturing semantic relationships between words. Moreover, previous research has not specifically explored public comments on political speeches on YouTube, which tend to be lengthy, informal, and code-mixed. To address this

* *Corresponding Author: Bella Risma Khailla Savana: bellarisma1407@gmail.com*

gap, this study applies FastText for sentiment analysis of public comments on Indonesian political speeches on YouTube from 2014 to 2024.

In addition to addressing challenges in Indonesian language processing, this research is also expected to contribute to the development of political sentiment analysis, both in terms of methodology and practical applications in the digital era.

2. LITERATURE REVIEW

2.1. Political Speech Comments

The analysis of comments on political speeches is an important approach to understanding how the public responds to messages delivered by national leaders. Comments appearing on social media platforms such as YouTube reflect various public reactions to political issues raised in speeches, ranging from support to criticism. Through sentiment analysis, researchers can classify public opinions into positive, negative, or neutral categories to evaluate the effectiveness of political communication (Medhat et al., 2014). In the digital era, the role of social media as a space for political discourse has become increasingly prominent, making user comments a rich source of data for studying public opinion (Liu, 2020).

However, analyzing comments on political speeches also presents complex linguistic challenges, particularly in the context of the Indonesian language. Social media comments often contain non-formal elements such as non-standard words, slang, code-mixing, irony, and sarcasm, all of which complicate the automatic classification process (Prabowo & Thelwall, 2009). Therefore, more advanced technical approaches such as word embedding and machine learning models are needed to capture the semantic nuances in text. These technologies help improve the accuracy of sentiment analysis in linguistically diverse and informal contexts such as YouTube comments.

2.2. Linguistic Characteristics of YouTube Comments

User comments on YouTube are often written in informal and highly varied language styles, reflecting the diversity of linguistic expression in society. In the context of sentiment analysis, this presents unique challenges, especially when comments use slang, local dialects, abbreviations, or even code-mixed language. The Indonesian language, in particular, has grammatical structures and writing styles that differ from English and often employs words with multiple or context-dependent meanings. These characteristics make it difficult for conventional sentiment analysis models not specifically trained on Indonesian data to perform accurately. Therefore, understanding the linguistic structure and local cultural context is essential in designing sentiment analysis systems that are more accurate and adaptive to the dynamics of language on social media (Prabowo & Thelwall, 2009).

2.3. Sentiment Analysis

Sentiment analysis is a technique within Natural Language Processing (NLP) that aims to identify and classify opinions or emotions in text into categories such as positive, negative, or neutral (Aziz, 2022) (Liu, 2020). This technique is widely used to analyze public opinion trends on emerging issues, particularly in social media contexts. In the case of political speeches, sentiment analysis helps understand public responses to statements or policies delivered by national leaders and evaluate the effectiveness of political communication (Prabowo & Thelwall, 2009).

Before classification, the comment data must go through a preprocessing stage to make it cleaner and more structured. This process includes removing special characters (data cleaning), separating concatenated words (word segmentation), normalizing non-standard words, and applying case folding to standardize text format. These steps are essential to

ensure that models such as FastText can optimally recognize sentiment patterns within the comments.

2.4. Sentiment Labeling

In this study, the labeling of comments was performed automatically using MDHugol, a pre-trained model based on IndoBERT, specifically designed for Indonesian-language sentiment classification. The model has been fine-tuned using sentiment datasets from Prosa AI, allowing it to effectively capture Indonesian linguistic patterns in determining text sentiment (Agung et al., 2024). MDHugol can classify text into three sentiment categories: positive, negative, and neutral. The model was accessed via Hugging Face and utilized without additional training (Agung et al., 2024). All comments were automatically labeled without manual correction, enabling a faster and more consistent labeling process.

2.5. FastText

FastText is a word embedding technique developed by Facebook AI Research (Bojanowski et al., 2017), which employs a subword approach by breaking words into character-level n-grams. This method allows the model to generate vector representations for rare or out-of-vocabulary (OOV) words, making it suitable for analyzing informal text such as YouTube comments. FastText also provides pre-trained models in various languages, including Indonesian, and constructs sentence representations by averaging word vectors. This approach supports efficient sentiment classification while effectively capturing the semantic meaning of text.

3. METHOD

3.1. Research Flow Diagaram

This study employs a quantitative approach using an experimental method, as it relies on numerical data and systematic model testing. YouTube comments that have been sentiment-labeled (positive, negative, and neutral) were processed using the FastText method. The model was then tested to classify sentiments, and its performance was evaluated using metrics such as accuracy, precision, recall, and F1-score.

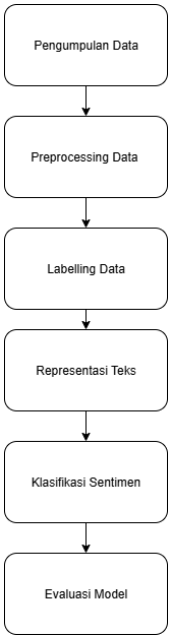

Figure 1. Research Flow Diagram

3.2. Data Collection

The data for this study were collected using a web scraping technique through the YouTube Data API v3, supported by Python scripts and libraries such as pytube and googleapiclient. A total of 12,102 comments were retrieved from political speech videos of President Joko Widodo uploaded on the Sekretariat Presiden and Sekretariat Kabinet RI YouTube channels between 2014–2024. The collected comments were stored in formats such as CSV or JSON and filtered based on their relevance to the research topic—Indonesian-language political speeches.

3.3. Data Preprocessing

Data preprocessing aims to clean and prepare textual data for analysis or modeling. The steps include:

3.3.1 Data Cleaning

Data cleaning removes irrelevant characters or elements such as URLs, hyperlinks, punctuation marks, special symbols, numbers, and emojis. This process ensures that the text data are clean and ready for further processing.

3.3.2 Word Segmentation

At this stage, merged or concatenated words are separated. A rule-based approach is applied using dictionary matching with an Indonesian word list. This process is conducted before labeling to ensure that the text fed into the labeling system is well-structured and aligns with proper Indonesian language syntax.

3.3.3 Case Folding

Case folding converts all text characters into lowercase to ensure consistency in text analysis.

3.4. Labelling Data

In this study, the mdhugol/Indonesian-bert-sentiment-classification model was used to automatically label the sentiment of comments. This transformer-based model was specifically trained to classify Indonesian-language sentiment into three categories: positive, neutral, and negative.

The final sentiment label for each comment was determined based on the class with the highest probability, which also indicates the model's confidence level. The labeling results were stored in a text format compatible with FastText for the next classification stage.

3.5. Text Representation

This study employed FastText both for text representation and sentiment classification. FastText uses a subword-based approach that breaks words into character n-grams, allowing it to effectively handle non-standard or newly formed words. Each comment was represented as the average of its constituent word vectors.

The model received labeled text input and classified it into three sentiment categories: positive, negative, and neutral. Training was conducted using pretrained Indonesian vectors to better capture the semantic meaning of words in context.

3.6. Sentiment Classification

Sentiment classification was performed to determine whether a comment expressed a positive, negative, or neutral sentiment using the FastText model. Comments were first

automatically labeled with the MDHugol IndoBERT-based model, then formatted as FastText input.

The dataset was split using the hold-out method (80% training data and 20% testing data) while maintaining label balance. The FastText model was trained using the train_supervised function with the following hyperparameters: Learning rate: 0.1, Vector dimension: 100, Word n-grams: 2, Epochs: 10.

The trained model was saved in .bin format for prediction purposes. Evaluation was performed on the test data by comparing predicted and true labels to compute accuracy and F1-score as key performance measures.

3.7. Model Evaluation

Model evaluation aimed to measure the performance of the FastText sentiment classification model and compare it with a traditional classification method using TF-IDF and Logistic Regression. The primary goal of this evaluation was to assess how well both models classified YouTube comments into three sentiment categories: positive, negative, and neutral.

3.7.1 Metrik Evaluasi

Evaluation was conducted using a confusion matrix, with the calculation of accuracy, precision, recall, and F1-score to provide a comprehensive view of model performance in classifying data.

3.7.2 Perbandingan dengan Metode Tradisional

The FastText model was compared with a traditional approach that used TF-IDF feature representation combined with the Logistic Regression algorithm. Comments were converted into feature vectors using unigram and bigram TF-IDF, then processed by the Logistic Regression model for sentiment classification. This comparison aimed to measure the advantages of word-embedding-based methods over traditional feature representation techniques.

3.7.3 Validasi Model Menggunakan K-Fold Cross Validation

Validation was performed using K-Fold Cross Validation to ensure that model performance was not dependent on a single data split. Each fold was alternately used as validation and training data. The evaluation metrics from all folds were averaged to produce more stable and reliable results. This validation process was applied to both the FastText and TF-IDF + Logistic Regression approaches.

4. RESULTS AND DISCUSSION

4.1. Data Collection

The initial dataset consisted of 12,102 YouTube comments obtained through a web scraping process. After undergoing preprocessing steps such as removing empty or irrelevant comments, text normalization, and word segmentation, the total number of usable comments was reduced to 11,798. This reduction aimed to ensure the quality and consistency of the data used for model training and evaluation in sentiment analysis.

4.2. Preprocessing Results

Preprocessing was carried out to clean and normalize YouTube comments before using them for model training. This process included removing irrelevant elements such as URLs, symbols, numbers, and emojis, as well as normalizing non-standard words. The preprocessing

stages conducted in this study included data cleaning, word segmentation, data normalization, and case folding. These steps ensured that the text data were properly structured and ready for further analysis using the sentiment classification model.

Table 1. Processing results

Original Comments	Preprocessed Comments
Terimakasih pak jokowi. Semoga kehidupan di seluruh indonesia kembali normal aŸ ‡ðŸ ‡ðŸ ‡ðŸ ‡	terimakasih pak jokowi semoga kehidupan di seluruh indonesia kembali normal
Acogan tv mengucapkan semoga racyat indonesia akan tbh lebih sehat2slalu dan sejahtera amin tuhan yang mh esa melindungi kita semua. amin yarobal alamin.	acogan televisi mengucapkan semoga. rakyat indonesia akan tambah lebih sehat selalu dan sejahtera amin tuhan yang maha esa melindungi kita semua amin yarobal alamin
Syarat naik kereta hipus pk yaksin boster Jgn pkebsyarat bbas lgi kek dlu toh d kerta gk jubel orangnya dkit	syarat naik kereta hapus pak vaksin booster jangan pakai syarat bebas lagi seperti dulu toh di kereta tidak jubel orangnya sedikit

4.3. Sentiment Labeling Results

A total of 11,798 comments were automatically labeled using the mdhugol/indonesia-bert-sentiment-classification model. The model classified each comment into three categories—positive, negative, and neutral, along with a probability score indicating prediction confidence. The labeling results were stored in a CSV file containing the comment text, sentiment label, and confidence score, which were later used as training and testing data for the FastText classification stage. From this process, the initial sentiment distribution was as follows: positive (5,220 comments), negative (4,693 comments), and neutral (1,885 comments). This class imbalance was addressed using Random Oversampling on the neutral class before the text representation stage, ensuring a more balanced data proportion and enabling the model to achieve more optimal training performance.

Table 2. Sentiment Labelling Results

Comments	Label
Kami dari Indonesia timur mengasihi Pak Jokowi. Terima kasih karena sudah memperhatikan kami yang jauh. God bless Pak Jokowi.	Sentiment: Positive (Score: 0,9764)
Bapak Jokowi tetap presiden dan pemimpin terbaik bangsa ini. Tanggung jawab bapak sangat besar terhadap negara ini. Semoga bapak bisa pantau terus perkembangan negara ini baik pembangunan maupun keamanan seluruh tanah air.	Sentiment: Positive (Score: 0.9768)

Comments	Label
Kapolda kapolres yang menahan pendemo ketar-ketir nih sekarang Kemarin Komisi HAM meminta nggak dihiraukan, sekarang presiden yang meminta.	Sentiment: Negative (Score: 0.9948)

4.4 Text Representation

In this study, two approaches were used to represent the comments: FastText and TF-IDF. The FastText representation employed pretrained Indonesian word embeddings (cc.id.300.bin) with a vector dimension of 300 and an n-gram configuration of 2, producing fixed-dimensional vectors for each word. Meanwhile, TF-IDF generated representations based on word frequency weights within the corpus to measure the importance of each word in a document. Tables 3 and 4 below show examples of the representation results from each method.

Table 3. FastText Representation Vector

No	Word	dim_1	dim_2	dim_3
1	yang	0.06253958	0.047223546	0.0795961
2	dan	-0.02841749	0.034917686	-0.04145676
3	pak	-0014613199	0.009694806	-0.012599981
4	tidak	0.120999314	0.10898251	0.13913022
5	jokowi	0.026111783	0.041840166	0.032997526

Table 4. TF-IDF Representation Vector

No	Kata	TF/IDF Score
1	memerangi	0.372676
2	semangat presiden	0.362478
3	covid	0.252882
4	semangat	0.236027
5	sangat	0.207288

4.5 Training and Evaluation Results

After the text representation stage, two classification approaches were tested to analyze the sentiment of comments: TF-IDF + Logistic Regression and FastText. The evaluation was conducted under four scenarios, covering both pre- and post-oversampling conditions, to assess the impact of data balancing on model performance. The performance was measured using accuracy, precision, recall, and F1-score across the three sentiment classes (positive, negative, neutral).

4.5.1 Evaluasi Model TF-IDF + Logistic Regression

The TF-IDF + Logistic Regression model without oversampling achieved an accuracy of 74.11%. The model was able to recognize positive and negative comments fairly well but

struggled to identify neutral comments, as reflected by the F1-score of 0.27 and recall of 0.17 for the neutral class. This imbalance caused the model to be more inclined to classify comments into dominant.

Table 5. Evaluation Results of TF-IDF + Logistic Regression Model

Class	Precision	Recall	F1-Score
Negative	0,6831	0,873	0,7665
Netral	0,7317	0,1676	0,2727
Positive	0,8085	0,817	0,8127
Accuracy			0,7411

After applying oversampling to the neutral class, the data distribution became more balanced, and the model's performance on minority classes improved. The overall accuracy increased to 74.92%, and the F1-score for the neutral class rose significantly from 0.27 to 0.53, indicating better recognition of neutral comments.

Table 6. Evaluation Results of TF-IDF + Logistic Regression Model After Oversampling

Class	Precision	Recall	F1-Score
Negative	0,7711	0,7966	0,7837
Netral	0,4687	0,6154	0,5321
Positive	0,8804	0,7548	0,8128
Accuracy			0,7492

4.5.2 Evaluation of FastText Model

The FastText model without oversampling achieved an accuracy of 76.82%, with the best performance in the positive (F1-score 0.82) and negative (0.80) classes. However, the F1-score for the neutral class (0.52) indicated that the model still faced challenges in handling unbalanced data.

Table 7. Evaluation Results of FastText Model

Class	Precision	Recall	F1-Score
Negative	0,7674	0,836	0,8002
Netral	0,5487	0,4934	0,5196
Positive	0,8437	0,8065	0,8247
Accuracy			0,7682

To address this imbalance, oversampling was applied to the neutral class before retraining the FastText model. The results showed improved balance among classes, although the total accuracy slightly decreased to 76.61%. The F1-score for the neutral class increased to 0.53, showing better recognition of neutral comments.

Table 8. Evaluation Results of FastText Model After Oversampling

Class	Precision	Recall	F1-Score
Negative	0,7711	0,8253	0,7973
Netral	0,5348	0,5305	0,5426
Positive	0,8491	0,7979	0,8227
Accuracy			0,7661

4.5.3 Comparative Analysis and Misclassification

Overall, FastText outperformed TF-IDF + Logistic Regression, with a margin of around 2–3% in accuracy. Although the difference was small, FastText proved to be more stable in recognizing informal language variations. The application of oversampling also helped balance class predictions, particularly improving the F1-score for the neutral class.

Some misclassifications were still found in comments containing irony, code-mixing, or implicit expressions. For example, sarcastic remarks detected as positive. This indicates that both models still face challenges in understanding semantic context in informal Indonesian text.

Table 9. Examples of Misclassification Analysis Results

No	Comment	Actual Label	Prediction	Explanation
1	*wan abud mencari kesempatan dalam penderitaan*	Negative	Netral	The model failed to recognize the subtle tone of criticism: the metaphorical phrase "seeking opportunity in suffering" was not identified as a negative sentiment.
2	*hikhik komen positif dapat nasbox*	Positive	Negative	The model failed to capture the tone of humor and mild sarcasm; the word "nasbox" is uncommon in the corpus, thus interpreted as having a negative connotation.
3	*almarhum bung karno tidur dengan tenang karna gagasan almarhum sudah terbayar lewat kerja keras pak jokowi*	Netral	Positive	This reflective sentence was interpreted as praise toward the president, even though it does not contain any explicit sentiment expression.
4	*sedih juga presiden yang suka kerja keras untuk rakyat harus selesal oktober*	Negative	Positive	The model struggled to distinguish the emotional expression "sedib" (a misspelling of "sedih", meaning sad), which conveys a negative tone, from the sentence context that praises the president.
5	*semoga bisa menyambungkan kereta cepat dari sumatra sampai papua*	Netral	Positive	The model interpreted the word "semoga" (hopefully) as an expression of optimism, thus classifying a neutral comment as positive.

Based on Table 4.9, many prediction errors occurred in comments with implicit contexts, sarcastic tones, or indirect expressions. The FastText model tended to be overly sensitive to positively connoted words such as "semoga" (hopefully) and "kerja keras" (hard work), even when the context was neutral or ambiguous. Moreover, the use of automatic labeling contributed to data noise, affecting prediction accuracy. This finding highlights the need for additional evaluation using manually annotated data to enhance the validity of results.

4.5.4 K-Fold Cross Validation Results

The validation process employed 5-fold cross-validation to assess the stability of model performance. The results showed that FastText consistently achieved higher accuracy and F1-scores than the TF-IDF + Logistic Regression method, particularly in recognizing neutral comments. The average accuracy of FastText reached 76.14%, slightly higher than 75.95% for TF-IDF + LR. The most significant improvement was observed in the neutral class F1-score, where FastText achieved an average of 0.4767, compared to 0.3522 for TF-IDF + LR.

Table 10. K-Fold Cross Validation Results

Model	Accuracy	F1 Positive Class	F1 Negative Class	F1 Netral Class
FastText	0.7614	0.8248	0.7942	0.4767
TF-IDF + LR	0.7595	0.8231	0.7919	0.3522

5. CONCLUSION

This study analyzed public sentiment toward the Indonesian President's political speech on YouTube using two approaches: TF-IDF + Logistic Regression and FastText. The results show that FastText achieved the highest accuracy (76.14%), slightly outperforming TF-IDF + Logistic Regression (75.95%). The use of oversampling effectively balanced class distribution and improved the model's ability to identify neutral comments.

The findings indicate that FastText is more effective in handling informal language, slang, and code-mixing, which are common in Indonesian social media comments. Nevertheless, traditional methods like TF-IDF remain relevant as strong baselines due to their competitive performance.

This study is limited by the number of comparative models and the absence of manual annotation in the labeling process. Future work could incorporate partial manual annotation and expand the variety of comparative models to enhance generalizability and sentiment analysis accuracy.

6. ACKNOWLEDGMENT

The authors would like to express their gratitude to the academic advisors and all individuals who provided guidance, feedback, and support throughout the preparation of this article. Appreciation is also extended to those who contributed to data collection and the implementation of this research.

7. REFERENCES

Agung, P., Wijaya, A., Dika, I. M., Hendra, I. K., & Jaya, T. (2024). Comprehensive analysis of teacher teaching performance through sentiment and POS tagging. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, 12(3), 147–156.

Alfariqi, F., Maharani, W., & Husen, J. H. (2020). Klasifikasi sentimen pada Twitter dalam membantu pemilihan kandidat karyawan dengan menggunakan convolutional neural network dan FastText embeddings. *E-Proceeding of Engineering*, 7(2), 8052–8062.

Aziz, A. (2022). Analisis sentimen identifikasi opini terhadap produk, layanan dan kebijakan perusahaan menggunakan algoritma TF-IDF dan SentiStrength. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 6(1), 115.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. https://doi.org/10.1162/tacl_a_00051

Pangestu, A. F., Rahmat, B., & Sihananto, A. N. (2024). Analisis sentimen pada media sosial X terhadap implementasi kurikulum merdeka menggunakan metode FastText dan long short-term memory (LSTM). *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, 9(4), 2271–2280.

Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions* (2nd ed.). Cambridge University Press. https://doi.org/10.1017/9781108639286

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2), 143–157. https://doi.org/10.1016/j.joi.2009.01.003

Putri, F. I., Wibawa, A. P., & Collante, L. H. (2024). Refining the performance of Indonesian-Javanese bilingual neural machine translation using Adam optimizer. *Jurnal Ilmiah ILKOM*, 16(3), 271–282. https://doi.org/10.33096/ilkom.v16i3.2467

Restya, B., & Cahyono, N. (2025). Optimasi metode klasifikasi menggunakan FastText dan Grid Search pada analisis sentimen ulasan Aplikasi SeaBank. *Jurnal Ilmiah Komunikasi & Informatika (JIKO)*, 9(1), 226–238. https://doi.org/10.26798/jiko.v9i1.1523

Speer, R., & Chin, J. (2016). An ensemble method to produce high-quality word embeddings. *arXiv preprint arXiv:1604.01692*. http://arxiv.org/abs/1604.01692