InsBERT: Word Importance from Artificial Insertions

Adam Osuský, Dávid Javorský and Ondřej Bojar

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Malostranské nám. 25, Prague, 118 00, Czech Republic

Abstract

We investigate the quantification of word importance by introducing a novel self-supervised task that modifies masked language modeling. Instead of predicting masked words, our approach involves learning to identify which words were inserted. We hypothesize that resulting models will predict a higher likelihood of insertion for less important words. We experiment with two different insertion strategies: the List Inserting Method (LIM) and the BERT Inserting Method (BIM). We outline the process for gathering manually estimated word importance data and describe the construction of a dataset for evaluating our methods. Our results indicate that our modified language modeling surpasses baselines and is competitive with existing research in assessing word importance.

Keywords

word importance, self-supervision, word insertion, synthetic data

1. Introduction

A significant amount of human knowledge and communication is now recorded as digital text [1, 2], often in raw, unstructured form [3]. This necessitates methods to make text searchable and easily summarized, driving the NLP community's interest in quantifying word importance as a possible solution.

The concept of identifying important words dates back to the 1950s [4], with early methods based on word frequencies such as TF-IDF , which remains widely used in modern NLP applications [5]. Various methods have been explored for determining word importance in tasks such as querying [6], summarization [7], text classification [8], and keyphrase extraction [9, 10].

Current approaches for assessing word importance involve comparing spatial distribution of words in the original versus shuffled text [11], exploiting attribution methods [12], utilizing χ^2 test [13, 14] or interpreting attention in attention-based models [15], although their interpretability is debated [16].

Kafle and Huenerfauth [17] collect annotations of word importance as real numbers from 0 to 1, which they later use for captioning to aid those who are deaf or hard of hearing [18].

Interestingly, [19] defines word importance ranks as the difference in the classifier's confidence for the target label when a specific word is included in the text versus when it is removed. This approach reveals that adversarial attack algorithms in NLP primarily disrupt the distribution of this word importance.

In [12], a method is explored to derive word significance from models trained for Natural Language Inference (NLI) and Paraphrase Identification (PI) by using an attribution method to compute scores for each input word, identifying those that contribute most to the model's decision. The approach involves training an interpreter to mask as many words as possible while still preserving the original prediction. We compare the performance of our approach with this work.

This study explores assessing word importance comprehensively, from collecting data to creating and evaluating an automatic word importance scorer. More precisely, the contribution of this work is: (1) A precise definition of word importance and proposed metrics for its evaluation, (2) a small multi-domain word-importance dataset in English annotated by three annotators, (3) a novel self-supervised machine learning method for predicting word importance. This self-supervised approach modifies BERT's [20] methodology to predict artificially inserted words rather than masked ones, examining two insertion methods: the List Inserting Method (LIM; inserting randomly from a word list) and the BERT Inserting Method (BIM; inserting using a BERT model). The results seem to indicate that our proposed method is superior to baselines such as TF-IDF and is on par or even better than existing approaches of calculating word importance.[1]

2. Word Importance

Word importance (WI) depends on its intended usage. Depending on objectives, such as text summarization or

[1] https://github.com/adam-osusky/predicting-word-importance

ITAT'24: Information technologies – Applications and Theory, September 20–24, 2024, Drienica, Slovakia

adam.osusky746@student.cuni.cz (A. Osuský); javorsky@ufal.mff.cuni.cz (D. Javorský); bojar@ufal.mff.cuni.cz (O. Bojar)

https://ufal.mff.cuni.cz/david-javorsky (D. Javorský); https://ufal.mff.cuni.cz/ondrej-bojar (O. Bojar)

0000-0003-2516-2535 (D. Javorský); 0000-0002-0606-0050 (O. Bojar)

grammar correction, the same words may hold different degrees of importance. In this work, we focus on semantic importance and we define it by drawing inspiration from prior works: Kafle and Huenerfauth [17] emphasize the loss incurred by removing a word and Javorský et al. [12] focus on the meaning contribution added by a word. We combine these perspectives into a unified definition of WI:

> Importance is the measure of a word's contribution to the overall meaning of the context, indicating the extent to which the removal of a word would diminish the information conveyed by the context.

We aim to collect human-annotated data for word importance, and therefore we need to clearly formulate instructions for annotators. Even though the most intuitive approach would be to let annotators score each word by a real number within the range $[0, 1]$, as it is done in other studies [17, 12], we find such task very difficult for annotators. Therefore, we represent WI as an importance *ranking*.

By *ranking*, we mean the ordering of word positions within a context, where the word position ranked as 1 is considered the most important. We rank word positions because the same word can appear multiple times in a context with varying levels of importance.

In our initial experiments, we observed that when annotators were unrestricted in the number of words they could rank, they tended to sequentially select key nouns from the subjects and objects, as well as the verbs that connect these elements in the sentences. However, this behavior does not align with our objectives.

Our primary goal is to identify the most important words, so the *ranking* can include only a subset of the word positions in a given context. For a context with the length of m, we aim to rank n positions where $1 \leq n \leq \lceil 0.1 \cdot m \rceil$. Word positions that are not ranked are assigned the rank of $n + 1$, referred as the "last rank". We term this process as rank limit being equal to 10%.

We argue that it might facilitate the annotators' attention on identifying only the most essential words in a given context. By restricting the number of words that can be ranked to 10% of the total word positions, we force a more selective process, making the annotators focus on the most salient words and not to be overwhelmed by numerous options of possible rankings.

Instructions for Annotators The full set of instructions provided to our annotators is as follows:

1. Arrange the words in descending order by their importance. You can rank at most 10 percent of the words, or choose to rank fewer if desired.
2. Create an order for the most important ones; any unranked words will receive the last rank, and they should be considered to have similar importance. At least one word must be ranked.
3. Click on words to select them. The selection order determines their ranking. Clicking on a selected word will deselect it. The first selected word is the most important.
4. Importance is the measure of a word's contribution to the overall meaning of the context. Indicating the extent to which the removal of a word would diminish the information conveyed by the context.
5. Contexts span five diverse domains: news, beletry, poetry, jokes, and transcribed spoken language.
6. In the transcribed spoken language domain, words may take the form of "(PERSON#NUMBER)" at the beginning of a person's reply, indicating the speaker's identity. These tags are non-clickable and non-rankable. Additionally, words in the form "PERSON#NUMBER" serve as references to other persons' names within the utterance.

A simple annotation tool was used for data collection. This tool allows annotators to rank words by clicking on them in sequence. If an annotator wants to insert a word into the middle of an already selected ranking, they must unselect the subsequent words and then reselect them in the desired order. While it might seem more convenient to allow direct insertion of a word into the middle of the ranking, the current approach has its benefits. By requiring annotators to reassess the subsequent rankings when making changes, the process encourages a more thoughtful and deliberate evaluation of the overall ranking.

3. Data Collection

To ensure diversity, we target various domains and their corresponding English datasets: News, the News Commentary dataset [21]; literature, data from [22]; poetry, data from [23]; jokes, data from [24]; and meeting transcripts, the ELITR Minuting Corpus [25]. From each domain, we manually select 10 contexts (each context possibly containing more sentences), ensuring that the contexts are around 60 words long. To achieve better granularity for certain words like "don't" and "I'm", the contexts are tokenized using the Moses tokenizer [26]. The dataset statistics are outlined in Table 1.

Each of the 50 contexts is annotated by three annotators who are non-native English speakers. These contexts with annotations form the Word Importance Dataset (WIDS). We make the dataset available at [27].

		Characters		Moses-tokens	
Domain	Contexts	Count	mean±std	Count	mean±std
News	10	2565	256.5±26.1	529	52.9±4.0
Literature	10	2207	220.7±17.1	601	60.1±7.2
Poetry	10	1776	177.6±27.5	540	54.0±6.2
Jokes	10	1938	193.8±25.4	575	57.5±7.0
Transcripts	10	2432	243.2±26.6	616	61.6±7.1
All	50	10918	218.4±38.5	2861	57.2±7.2

Table 1
Statistics of our Word Importance Dataset. The mean and standard deviation are computed on lengths of contexts.

Domain	Pair1-2	Pair1-3	Pair2-3	Average
News	0.318	0.296	0.388	0.334
Literature	0.223	0.286	0.273	0.260
Poetry	0.260	0.332	0.238	0.277
Jokes	0.533	**0.630**	**0.533**	**0.565**
Transcripts	**0.539**	0.475	0.518	0.511
All	0.380	0.406	0.395	0.393

Table 2
Cohen's kappa coefficient between pairs of annotators within individual domains and across all domains in our Word Importance Dataset. The highest agreements within each domain are highlighted in bold.

To assess the similarity of the annotations, we compute inter-annotator agreement using Cohen's kappa [28]. We simplify the calculation by classifying each word position in every context as either "selected" or "not selected" by annotators. In Section 5.2, we present metrics to incorporate the order of selection for a more nuanced analysis.

The computed Cohen's kappa values are shown in Table 2. It is unsurprising that one out of the two domains with the least agreement than poetry. An intriguing observation is that literature displays slightly lower agreement as poetry. The domains with the highest agreement are jokes and meeting transcripts. We find these findings in line with our intuition: There is often very clear what words make jokes funny and speech in meetings may contain many objectively unimportant words, e.g. filler words, hesitations, false starts etc.

4. Methodology

Our approach involves fine-tuning a pre-trained BERT model [20] using automatically generated data. Specifically, we generate training text data by inserting words into existing text and then use the modified text as training data. The objective of fine-tuning is to predict which words were inserted. We hypothesize that this task will require the model to understand the importance of each word and its contribution to the overall meaning of the context, ultimately leading the model to assign higher likelihoods of insertion to less important words. This enables us to create a *ranking* of words in a test input text by ordering them according to their predicted word importance score. We propose two methods for creating the training dataset of inserted words: the *List Inserting Method* (LIM) and the *BERT Inserting Method* (BIM).

List Inserting Method (LIM) This method inserts words randomly from a predefined list. This list is generated by splitting the base corpus into words by white space. Consequently, words that appear more frequently in the corpus are more likely to be inserted, mirroring the original distribution.

BERT Inserting Method (BIM) This method aims to insert words that do not fit well in the sentence. This is achieved by leveraging the capabilities of another instance of the BERT model [20].[2] Because BERT predicts the words without any information except the text itself, we assume that they should not alter the sentence's meaning significantly. In this method, mask tokens are placed at the selected insertion positions within the text and BERT is then used to predict the masked tokens. We prohibit predictions of neighboring tokens (those immediately before and after the masked token) and sub-word tokens, i.e. tokens that are not a beginning of a word. After filtering out these unwanted tokens, we select the prediction with the highest logit probability.

For both methods, possible positions for insertion include places before existing words and one additional position at the end of the text. These words are obtained by splitting the text by white space to determine the potential insertion positions. We insert at most one word in each position, ensuring that words are not inserted consecutively.

The positions for insertion are selected randomly. The insertion rate is defined as the ratio of the number of words to be inserted to the total number of words in the original text sample. For instance, if a text sample contains 10 words and we use an insertion rate of 0.5, we insert 5 new words into this text.

In our experiments, our goal is to effectively compare the two insertion methods, LIM and BIM, as well as evalu-

[2] https://huggingface.co/google-bert/bert-base-uncased

In the 1970s, failures. Wheeler a became increasingly the forgetful and Olympic came to rely largely dwarf on his Seattle assistant, Molly rumored Myres, Poesaka to killed organise Police his Army affairs. souls Amid increasing ill about health, the in September 1973 pairs he moved in full-time into Mills, Myres's house in Byzantines. Leatherhead, time Surrey, although said he claim continued to use (RCA his EMS central Hill London "When flat during details day-trips Essendon. to the He city. There, he Morison authored a final was book, My Archaeological Mission who to India Studios and as Pakistan, largely although polygyny much for of the text was with culled as from his previous publications; the it was veterans published none by Thames was and Hudson in 1976. After Charles suffering that a stroke, on Wheeler went died The at On Myers' Religious home on Kierszenbaum 22 July eldest 1976. Archives' In bridge memoriam, the British Academy, Royal Academy, and Royal Society Ukraine flew their flags cites at half-mast. membership Wheeler's from funeral was player held with military review trappings in at a was local demolished. crematorium, while a contacting larger memorial service was held a in south St (April James's aid Church, area, Piccadilly in November.

late In the 1970s, " Wheeler became " increasingly forgetful and came personal to " rely personal largely " on his " assistant, late Molly back Myres, country to organise his " affairs. apartment Amid occasional increasing visit ill health, in later September 1973 east he moved full-time which into original Myres's two house in Leatherhead, Surrey, although he posthumously continued press to use shortly his central London major flat finally during " day-trips to " the prestigious city. There, birmingham he authored a final book, My bafta Archaeological other Mission to all India " and Pakistan, although much of " the state text was culled also from his ceremonial previous publications; " it nearby was published much by " Thames " and catholic Hudson in 1976. new After john suffering a back stroke, Wheeler (died at solely Myers' " home " on 22 July 1976. " In memoriam, the " British Academy, john Royal Academy, also and " Royal Society " flew their flags capital at half-mast. first Wheeler's research funeral was " held with military directly trappings what at a alone local crematorium, while a john larger george memorial service also was held in St James's michael Church, " Piccadilly in November.

Figure 1: Example of text with inserted words using LIM (left) and BIM (right) methods with insertion rate 0.5. Words that are highlighted in yellow boxes are inserted.

ate their performance across different insertion rates. To achieve this, we train separate models for both methods at insertion rates of 0.25, 0.5, and 0.75.

In future endeavors, it would be interesting to extend this research by training models on datasets created using both LIM and BIM, potentially combining or varying insertion rates.

4.1. Example Text with Inserted Words

In Figure 1, we illustrate an example of text from our preprocessed WikiText dataset (Section 5.1), where words have been inserted using both the LIM and BIM methods.

The inserted words in the BIM method often appear superfluous, adding information to the sentences. Notably, there is a frequent insertion of apostrophes, occurring more often than desired. To investigate this phenomenon further, we conducted a simple experiment on a subset of 100 examples to analyze how the frequency of apostrophes changes with varying insertion rates. Refer to Figure 2 for the results. It is observed that the frequency of apostrophes converges to approximately 0.22 when the insertion rate is at least 0.5.

Conversely, in the LIM method, the inserted words sometimes introduce information that seems out of context. Additionally, some inserted words include punctuation marks, such as "(April" or "area,", as seen in the last sentence on the left in Figure 2.

5. Experiments

5.1. Training Details

We detail the preprocessing methods applied to the WikiText dataset and outline the construction of the training regime.

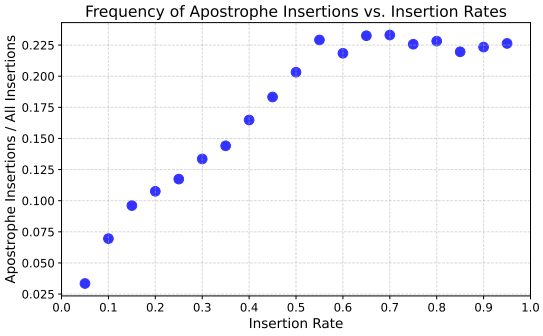

Figure 2: Frequency of apostrophes in inserted words by BIM method with varying insertion rates.

Data As the base text corpus in which we insert new words, we have selected the WikiText dataset [29]. This dataset comprises articles from Wikipedia[3] that are classified as either *Good* or *Featured* articles, according to the criteria specified by Wikipedia editors at the time of creation.

We use version *wikitext-103-raw-v1*[4] from the HuggingFace datasets library [2]. Each example in the dataset is either a paragraph or a title. For our specific use case, we preprocess this dataset by removing examples that are titles. Additionally, the dataset is tokenized using the Moses tokenizer [26]. Since we employ a Transformer [15] model that already incorporates its own tokenization, we detokenize the text. We retain the original train-validation-test splits. For detailed statistics regarding our preprocessed WikiText, refer to Table 3.

[3] https://www.wikipedia.org/
[4] https://huggingface.co/datasets/wikitext

	Train	Validation	Test
Paragraphs	859,532	1841	2183
Characters	509,512,733	1,083,136	1,217,025
Characters mean ± std	592.7 ± 404.2	588.3 ± 385.4	557.5 ± 404.1
Words	84,208,748	178,815	201,013
Words mean ± std	97.9 ± 67.1	97.1 ± 63.8	92.0 ± 67.3

Table 3
Statistics for train, validation, and test splits of our preprocessed WikiText. The statistics are computed on lengths of paragraphs, where std stands for empirical standard deviation. Words are obtained by splitting the detokenized text by white space.

Model	F_1	Loss	Precision	Recall
BIM-0.25	0.929	0.046	0.930	0.927
LIM-0.25	0.952	0.031	0.957	0.947
BIM-0.5	0.953	0.048	0.950	0.957
LIM-0.5	0.972	0.028	0.974	0.971
BIM-0.75	0.982	0.027	0.980	0.984
LIM-0.75	**0.985**	**0.021**	**0.985**	**0.985**

Table 4
Performance of models on the test split from their respective generated data. The model names indicate whether the List Inserting Method (LIM) or BERT Inserting Method (BIM) was used for generating data, along with the insertion rate. Loss refers to cross-entropy loss, and in the computation of F_1 score, precision, and recall, the positive target is the class for inserted words. Best results are in bold.

Models We create six datasets using both the LIM and BIM techniques, each with insertion rates of 0.25, 0.5, and 0.75. Subsequently, we train six models, each corresponding to a distinct combination of insertion method and rate.

To ensure a fair comparison and avoid introducing bias due to differences in hyper-parameters, we use identical settings for all models. Hyper-parameters were selected empirically based on initial experiments.

We use a learning rate of 0.0032, batch size of 256, Adam optimizer [30] with default betas $(0.9,0.999)$, and a linear learning rate scheduler with a linear warmup of 350 steps. Starting from the BERT[5] pre-trained model, we fine-tune on each of the datasets for 5 epochs.

The performance of individual models in the classification task is shown in Table 4. The LIM models consistently outperform the BIM models. Given this discrepancy and the distribution of inserted punctuation marks discussed in Section 4.1, it indicates that the BIM data present a more challenging task, as the inserted words blend more seamlessly with the context.

5.2. Evaluation for WI ranking

Our trained models predict logits for the probability of word insertion. We construct the *ranking* by ordering BERT-tokens in one context in ascending order of their insertion probabilities. Since Word Importance Dataset (WIDS) is pretokenized by the Moses tokenizer, we use the logits of the first BERT-token to score the original Moses-token if a Moses-token is split into multiple BERT-tokens.

For the human reference, we calculate the average rank of each token based on the *rankings* provided by all three annotators and then order the words according to these average ranks. With only three annotators, a majority of words still fall into the lowest rank, leading to inconsistencies between model predictions and averaged annotations, as they can result in different lowest ranks. To ensure consistency in evaluation, we apply the 10% rank limit to both the averaged annotations and model predictions.

Since 90% of the positions fall into the lowest rank, this creates challenges in designing effective evaluation metrics. To address these issues, we propose three metrics, each progressively refining and incorporating desired properties to better align with our evaluation goals.

Pearson correlation The simplest and well-known approach is to calculate the sample Pearson correlation coefficient on the ranks of word positions over all positions and all contexts in the dataset. However, this method is not ideal because 90% of word positions within a given context fall into the lowest rank. Our primary focus is on achieving higher agreement within the top 10%, which is not adequately emphasized by this correlation measure.

***k*-inter** Another perspective on *rankings* is to consider words that do not have the last rank and disregard their specific order. By doing this, we view the *rankings* as indicators of which words are important, allowing us to measure the extent of the intersection between different *rankings*. We thus propose a new metric, ***k*-inter**, where we filter the *ranking* and keep only word positions that do not have the last rank. We then compute the fraction of context pairs where the intersection of their filtered *rankings* has at least k elements. We examine values of $k \in \{1, 2, 3\}$.

[5] https://huggingface.co/google-bert/bert-base-uncased

Annotators	Pearson	1-inter	2-inter	3-inter	Overlap
Pair1-3	0.534	0.90	**0.76**	**0.52**	0.319
Pair1-2	0.555	**0.92**	0.72	0.38	0.324
Pair2-3	**0.602**	0.90	0.70	0.46	**0.394**
Average	0.563	0.91	0.73	0.45	0.346

Table 5
Metrics from Section 5.2 computed between our annotators on Word Importance Dataset.

Domain	Pair 1-2	Pair 1-3	Pair 2-3	Average
News	0.286	0.247	**0.511**	0.348
Literature	0.211	0.220	0.340	0.257
Poetry	0.189	0.301	0.220	0.237
Jokes	**0.484**	**0.475**	0.462	**0.474**
Transcripts	0.450	0.354	0.437	0.413

Table 6
Overlap computed between our annotators on Word Importance Dataset, but on individual domains.

Model	Pearson	1-inter	2-inter	3-inter	Overlap
Random	0.256	0.54	0.13	0.01	0.061
PI	0.321	0.78	0.40	0.08	0.114
TF-IDF	0.309	0.66	0.20	0.04	0.121
BIM-0.75	0.335	0.82	0.32	0.12	0.125
BIM-0.25	0.341	0.76	0.40	0.14	0.131
LIM-0.5	0.328	0.72	0.40	0.12	0.137
LIM-0.75	0.352	0.80	0.48	0.18	0.142
BIM-0.5	0.344	0.70	0.42	0.14	0.143
NLI	0.374	**0.90**	**0.56**	**0.22**	0.150
LIM-0.25	**0.376**	0.82	0.52	0.14	**0.178**
Humans	0.563	0.91	0.73	0.45	0.346

Table 7
Evaluation of models from Section 6 on the Word Importance Dataset. The "Random"category represents the average metrics of 100 random predictions, while "Humans" denotes the average of human metrics from Table 5. The metrics are defined in Section 5.2.

Overlap The limitation of k-inter is that it does not consider specific rank values, only if the words are in the top 10%. We aim to assign more weight to agreements on specific rank values, prioritizing the match on higher-ranked agreements over lower-ranked ones. We thus propose to use the *average overlap* metric, as described by Webber et al. [31]. First, we derive an ordered list of words from the *ranking*. The agreement between lists l and $\bar{l}$ at depth d is defined as $A(l, \bar{l}, d) = |l_{:d} \cap \bar{l}_{:d}|/d$, where $l_{:d}$ represents the first d elements of the list. The average overlap at depth k is then $AO(l, \bar{l}, k) = \frac{1}{k} \sum_{d=1}^{k} A(l, \bar{l}, d)$. For context pairs of *rankings*, we compute the average overlap for each pair and then average these values, which we refer to as **overlap**. The depth is chosen differently for each pair: for a context with length m, the depth is set to $\lceil 0.1 \cdot m \rceil$, to be consistent with our rank limit of 10%.

6. Results

We first evaluate the pair-wise agreement between annotators using these metrics, which we present in Table 5. This evaluation complements Cohen's kappa from Section 3. The order of annotator pairs remains consistent for both the Pearson correlation and the overlap metric. The k-inter values for k values of 1 and 2 are relatively high compared to Pearson correlation or the overlap, indicating that the annotators agree on the selection of the most important words but not that well on their order. This supports our decision to let annotators focus only on the most important words and not make them mentally overloaded by the vast amount of options. In Table 6, we further present the overlap between annotators within individual domains of WIDS. Annotators show the highest similarity in the jokes domain and the lowest in the poetry domain. This observation aligns with the results in Table 2. For other metrics on individual domains, see Appendix A.

Finally, we evaluate the performance of all six of our trained models. Additionally, we include random predictions as a baseline for our metrics and the average human performance from Table 5 as an upper bound.

As an additional baseline, we include term frequency–inverse document frequency (TF-IDF), computed on the Word Importance Dataset without any preprocessing. Furthermore, we include two models, PI (Paraphrase Identification) and NLI (Natural Language Inference), developed by Javorský et al. [12]. We obtain *rankings* from all models by ordering the words according to their significance scores.

The results are presented in Table 7, indicating that our models are performing reasonably well. They surpass random predictions and TF-IDF across all metrics and are comparable to the NLI model. Notably, LIM-0.25 even exceeds the NLI model in both the overlap and Pearson correlation metrics. Metrics that consider the order of selected words show our models are approximately halfway to achieving human-level performance. They are approaching human performance in terms of 1-inter but lag significantly in higher k-inter metrics.

It is quite surprising that LIM approach is superior to BIM, suggesting that simple methods are sometimes more efficient. We hypothesize that inserted words by BERT are so well suited to the surrounding context that it is very difficult to detect them, which effectively decreases the useful learning signal from them.

For readers interested in a detailed view of all metrics across individual domains, refer to Appendix A.

7. Conclusion

In this paper, we define word importance, collect annotations for a small multi-domain word-importance dataset in English, propose metrics for its evaluation and introduce a novel self-supervised machine learning method: The goal is to predict inserted words in the text. Our results demonstrate that our method outperforms baseline models and is comparable to prior work on word importance.

Possible future work might benefit from more experiments when using BIM or combining LIM and BIM, potentially leading to more competitive results. Experimenting with smaller insertion ratios can be another potential avenue.

Limitations One of the primary limitations of our study is the size of the Word Importance Dataset, since it includes only 50 relatively short contexts that consists of approximately 60 words. Varying lengths of context might contribute to better generalization. The study compares importance scores to only one other indicator of word significance and it also lacks the evaluation of importance scores on a downstream task.

Another limitation is the small number of annotators. With a larger pool of annotators, the data in the Word Importance Dataset would likely exhibit lower variance. This would result in higher quality averaged *rankings* that are more closely aligned with the true distribution.

Finally, the work does not provide the evaluation of importance scores on the word-importance dataset collected by Kafle and Huenerfauth [17].

Acknowledgments

The work has been partially supported by the grants 272323 of the Grant Agency of Charles University, 19-26934X (NEUREM3) of the Czech Science Foundation and SVV project number 260 698.

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] G. Penedo, H. Kydlíček, L. von Werra, T. Wolf, Fineweb, 2024. URL: https://huggingface.co/datasets/HuggingFaceFW/fineweb. doi:10.57967/hf/2092.

[2] Q. Lhoest, A. V. del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall, et al., Datasets: A community library for natural language processing, arXiv preprint arXiv:2109.02846 (2021).

[3] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, M. R. Yeganegi, Text mining in big data analytics, Big Data and Cognitive Computing 4 (2020) 1.

[4] H. P. Luhn, The automatic creation of literature abstracts, IBM Journal of research and development 2 (1958) 159–165.

[5] M. Das, P. Alphonse, et al., A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset, arXiv preprint arXiv:2308.04037 (2023).

[6] Z. Dai, J. Callan, Context-aware document term weighting for ad-hoc search, in: Proceedings of The Web Conference 2020, 2020, pp. 1897–1907.

[7] K. Hong, A. Nenkova, Improving the estimation of word importance for news multi-document summarization, in: S. Wintner, S. Goldwater, S. Riezler (Eds.), Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Gothenburg, Sweden, 2014, pp. 712–721. URL: https://aclanthology.org/E14-1075. doi:10.3115/v1/E14-1075.

[8] I. Sheikh, I. Illina, D. Fohr, G. Linarès, Learning word importance with the neural bag-of-words model, in: P. Blunsom, K. Cho, S. Cohen, E. Grefenstette, K. M. Hermann, L. Rimell, J. Weston, S. W.-t. Yih (Eds.), Proceedings of the 1st Workshop on Representation Learning for NLP, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 222–229. URL: https://aclanthology.org/W16-1626. doi:10.18653/v1/W16-1626.

[9] M. Song, Y. Feng, L. Jing, A survey on recent advances in keyphrase extraction from pre-trained language models, Findings of the Association for Computational Linguistics: EACL 2023 (2023) 2153–2164.

[10] B. Xie, J. Song, L. Shao, S. Wu, X. Wei, B. Yang, H. Lin, J. Xie, J. Su, From statistical methods to deep learning, automatic keyphrase prediction: A survey, Information Processing & Management 60 (2023) 103382.

[11] A. Mehri, M. Jamaati, H. Mehri, Word ranking in a single document by jensen–shannon divergence, Physics Letters A 379 (2015) 1627–1632.

[12] D. Javorský, O. Bojar, F. Yvon, Assessing word importance using models trained for semantic tasks, 2023. arXiv:2305.19689.

[13] X. Li, X. Wu, X. Hu, F. Xie, Z. Jiang, Keyword extraction based on lexical chains and word co-occurrence for chinese news web pages, in: 2008 IEEE International Conference on Data Mining Workshops, IEEE, 2008, pp. 744–751.

[14] H. Jiao, Q. Liu, H.-b. Jia, Chinese keyword extraction based on n-gram and word co-occurrence, in: 2007 International Conference on Computational

Intelligence and Security Workshops (CISW 2007), IEEE, 2007, pp. 152–155.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[16] S. Serrano, N. A. Smith, Is attention interpretable?, arXiv preprint arXiv:1906.03731 (2019).

[17] S. Kafle, M. Huenerfauth, A corpus for modeling word importance in spoken dialogue transcripts, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://aclanthology.org/L18-1016.

[18] S. Kafle, P. Yeung, M. Huenerfauth, Evaluating the benefit of highlighting key words in captions for people who are deaf or hard of hearing, in: Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility, 2019, pp. 43–55.

[19] L. Shen, X. Zhang, S. Ji, Y. Pu, C. Ge, X. Yang, Y. Feng, Textdefense: Adversarial text detection based on word importance entropy, arXiv preprint arXiv:2302.05892 (2023).

[20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[21] J. Tiedemann, Parallel data, tools and interfaces in OPUS, in: N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey, 2012, pp. 2214–2218. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

[22] R. Nagyfi, Dataset card for project gutenber - english language ebooks, https://huggingface.co/datasets/sedthh/gutenberg_english, 2023. Accessed: 2024-03-28.

[23] A. Parrish, Github repository for gutenberg-poetry-corpus, https://github.com/aparrish/gutenberg-poetry-corpus, 2018. Accessed: 2024-03-28.

[24] SocialGrep, Dataset card for one-million-reddit-jokes, https://huggingface.co/datasets/SocialGrep/one-million-reddit-jokes, 2021. Accessed: 2024-03-28.

[25] A. Nedoluzhko, M. Singh, M. Hledíková, a. T. Ghosal, O. Bojar, Elitr minuting corpus: A novel dataset for automatic minuting from multi-party meetings in english and czech, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 3174–3182.

[26] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst, Moses: Open source toolkit for statistical machine translation, in: S. Ananiadou (Ed.), Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 177–180. URL: https://aclanthology.org/P07-2045.

[27] A. Osuský, D. Javorský, Word importance dataset, 2024. URL: http://hdl.handle.net/11234/1-5520, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[28] J. Cohen, A coefficient of agreement for nominal scales, Educational and psychological measurement 20 (1960) 37–46.

[29] S. Merity, C. Xiong, J. Bradbury, R. Socher, Pointer sentinel mixture models, 2016. arXiv:1609.07843.

[30] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[31] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Transactions on Information Systems (TOIS) 28 (2010) 1–38.

A. Metrics on Individual Domains

In Table 9, we present our proposed metrics computed for individual domains within the Word Importance Dataset between human *rankings*. Notably, the poetry domain exhibits relatively high k-inter values, whereas the Pearson correlation and overlap metrics are low. This indicates that humans agreed more on which words are important rather than on the order of their importance.

In Table 8, we present the overlap of the models from Section 6 across individual domains within the WIDS. Our models outperform the TF-IDF baseline in all domains except for the news domain. In a few cases and metrics, Random ranking outperforms some methods. It is worth noting that each domain includes only 10 examples, which may lead to significant variability in the results. Despite this, human performance consistently exceeds that of the models across all domains.

In Table 10, we present all of our proposed metrics computed for models from Section 6 on individual domains within the Word Importance Dataset. For these evaluations, the TF-IDF was created using text solely from the respective individual domain. It is apparent that the performance ordering of the models is not consistent across the different domains, likely due to each domain having only 10 examples.

An interesting observation is that TF-IDF performs best on the news domain, whereas it is under performing in the other domains.

Model	News	Lit.	Poetry	Jokes	Trans.
Random	0.066	0.061	0.062	0.068	0.065
PI	0.068	0.185	0.120	0.106	0.095
TF-IDF	**0.126**	0.055	0.044	0.088	0.142
BIM-0.75	0.075	0.196	0.121	0.106	0.125
BIM-0.25	0.069	0.212	0.078	0.130	0.166
LIM-0.5	0.063	0.156	0.158	0.143	0.167
LIM-0.75	0.048	0.128	0.135	**0.192**	0.206
BIM-0.5	0.079	0.170	0.114	0.084	0.270
NLI	0.047	**0.221**	**0.207**	0.153	0.120
LIM-0.25	0.115	0.133	0.159	0.183	**0.299**
Humans	0.348	0.257	0.237	0.474	0.413

Table 8
Overlap of models from Section 6 on the Word Importance Dataset for individual domains. The "Random" category represents the average metrics of 100 random predictions, while "Humans" denotes the average of human metrics from Table 6. Sorted according to overlap score across the domains (not shown here).

Annotators	Pearson	1-inter	2-inter	3-inter	4-inter	5-inter	Overlap
News							
pair1-3	0.378	0.90	0.60	0.40	0.00	0.00	0.247
pair1-2	0.412	0.90	0.60	0.30	0.00	0.00	0.286
pair2-3	0.631	1.00	0.70	0.30	0.10	0.00	0.511
Average	0.474	0.93	0.63	0.33	0.03	0.00	0.348
Literature							
pair1-2	0.472	0.80	0.50	0.20	0.00	0.00	0.211
pair1-3	0.440	0.70	0.60	0.40	0.20	0.10	0.220
pair2-3	0.535	0.80	0.50	0.30	0.10	0.00	0.340
Average	0.483	0.77	0.53	0.30	0.10	0.03	0.257
Poetry							
pair1-2	0.413	0.90	0.60	0.10	0.00	0.00	0.189
pair2-3	0.422	0.70	0.50	0.20	0.10	0.00	0.220
pair1-3	0.481	0.90	0.70	0.30	0.30	0.00	0.301
Average	0.439	0.83	0.60	0.20	0.13	0.00	0.237
Jokes							
pair2-3	0.597	1.00	1.00	0.80	0.40	0.10	0.462
pair1-3	0.641	1.00	1.00	0.90	0.50	0.30	0.475
pair1-2	0.614	1.00	1.00	0.60	0.50	0.20	0.484
Average	0.617	1.00	1.00	0.77	0.47	0.20	0.474
Transcripts							
pair1-3	0.565	1.00	0.90	0.60	0.50	0.10	0.354
pair2-3	0.683	1.00	0.80	0.70	0.40	0.00	0.437
pair1-2	0.671	1.00	0.90	0.70	0.30	0.10	0.450
Average	0.640	1.00	0.87	0.67	0.40	0.07	0.413

Table 9
Metrics from Section 5.2 computed between our annotators on individual domains from the Word Importance Dataset corpus.

Model	Pearson	1-inter	2-inter	3-inter	4-inter	5-inter	Overlap
News							
NLI	0.17	0.80	0.30	0.00	0.00	0.00	0.047
LIM-0.75	0.168	0.60	0.10	0.00	0.00	0.00	0.048
LIM-0.5	0.177	0.40	0.30	0.10	0.00	0.00	0.063
Random	0.178	0.52	0.12	0.01	0.00	0.00	0.066
PI	0.189	0.80	0.20	0.00	0.00	0.00	0.068
BIM-0.25	0.191	0.50	0.10	0.10	0.00	0.00	0.069
BIM-0.75	0.216	0.70	0.20	0.00	0.00	0.00	0.075
BIM-0.5	0.183	0.30	0.20	0.10	0.00	0.00	0.079
LIM-0.25	0.235	0.70	0.40	0.00	0.00	0.00	0.115
tf-idf	0.229	0.60	0.30	0.00	0.00	0.00	0.126
Literature							
tf-idf	0.231	0.60	0.20	0.00	0.00	0.00	0.055
Random	0.251	0.55	0.15	0.02	0.00	0.00	0.061
LIM-0.75	0.292	0.80	0.40	0.10	0.00	0.00	0.128
LIM-0.25	0.302	0.70	0.40	0.00	0.00	0.00	0.133
LIM-0.5	0.302	0.80	0.20	0.10	0.00	0.00	0.156
BIM-0.5	0.345	0.90	0.50	0.10	0.00	0.00	0.170
PI	0.34	0.70	0.50	0.10	0.10	0.00	0.185
BIM-0.75	0.354	1.00	0.30	0.10	0.00	0.00	0.196
BIM-0.25	0.379	0.90	0.50	0.10	0.00	0.00	0.212
NLI	0.438	0.90	0.70	0.30	0.10	0.00	0.221
Poetry							
tf-idf	0.236	0.50	0.10	0.00	0.00	0.00	0.044
Random	0.255	0.52	0.13	0.01	0.00	0.00	0.062
BIM-0.25	0.279	0.80	0.40	0.00	0.00	0.00	0.078
BIM-0.5	0.293	0.70	0.50	0.00	0.00	0.00	0.114
PI	0.364	0.90	0.60	0.20	0.00	0.00	0.120
BIM-0.75	0.356	0.80	0.50	0.10	0.00	0.00	0.121
LIM-0.75	0.35	0.80	0.60	0.20	0.00	0.00	0.135
LIM-0.5	0.363	0.70	0.60	0.20	0.10	0.00	0.158
LIM-0.25	0.357	0.90	0.40	0.10	0.10	0.10	0.159
NLI	0.435	0.90	0.70	0.50	0.00	0.00	0.207
Jokes							
Random	0.179	0.55	0.16	0.02	0.00	0.00	0.068
BIM-0.5	0.214	0.70	0.30	0.00	0.00	0.00	0.084
tf-idf	0.203	0.60	0.30	0.10	0.00	0.00	0.088
PI	0.264	0.80	0.40	0.10	0.10	0.00	0.106
BIM-0.75	0.238	0.80	0.20	0.10	0.00	0.00	0.106
BIM-0.25	0.269	0.70	0.40	0.20	0.00	0.00	0.130
LIM-0.5	0.253	0.70	0.30	0.10	0.00	0.00	0.143
NLI	0.325	1.00	0.50	0.10	0.10	0.10	0.153
LIM-0.25	0.327	0.80	0.80	0.20	0.00	0.00	0.183
LIM-0.75	0.312	1.00	0.60	0.10	0.00	0.00	0.192
Transcripts							
Random	0.213	0.55	0.14	0.01	0.00	0.00	0.065
PI	0.248	0.70	0.30	0.00	0.00	0.00	0.095
NLI	0.294	0.90	0.60	0.20	0.00	0.00	0.120
BIM-0.75	0.301	0.80	0.30	0.30	0.10	0.00	0.125
tf-idf	0.309	0.90	0.30	0.00	0.00	0.00	0.142
BIM-0.25	0.355	0.90	0.60	0.30	0.10	0.10	0.166
LIM-0.5	0.33	1.00	0.60	0.10	0.10	0.00	0.167
LIM-0.75	0.409	0.80	0.70	0.50	0.20	0.00	0.206
BIM-0.5	0.438	0.90	0.60	0.50	0.10	0.10	0.270
LIM-0.25	0.446	1.00	0.60	0.40	0.00	0.00	0.299

Table 10
Metrics from Section 5.2 computed for models from Section 6 on individual domains from the Word Importance Dataset. Sorted by overlap within each domain.