Probing BERT for Animacy Understanding in Semantic Noun Classes

Aldi Halili

Center for Information and Language Processing, LMU Munich
Aldi.Halili@campus.lmu.de

August 27, 2024

Abstract

We evaluate the BERT model's capability based on its contextual word embeddings, with a particular focus on capturing semantic information related to the animacy of nouns. This investigation explores the extent of BERT's knowledge regarding animate and inanimate objects. Using the Natural Language Toolkit (NLTK) and WordNet ontology, we annotated 45,427 nouns as either animate or inanimate. A Multilayer Perceptron (MLP) classifier, employed as a probing mechanism, was then trained on this data. The results indicate that BERT struggles to effectively distinguish between animate and inanimate nouns, as reflected by the near-identical cosine similarity scores for both categories. Additionally, WordNet annotations were not validated, and frequent misclassifications were observed, making WordNet a less reliable tool for animacy annotation. This study highlights the limitations of using WordNet and BERT for animacy detection and suggests that future work should involve validation with gold standard data to improve results.

1 Introduction

In recent years, pre-trained models like ELMo (Embeddings from Language Models) proposed by Peters et al. (2018a) and BERT (Bidirectional Encoder Representations from Transformers) developed by Devlin et al. (2018) have significantly improved performance across a wide range of tasks in the field of Natural Language Processing (NLP). The success of these models lies in their contextualized word embeddings, which are increasingly replacing static word embeddings Models like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Despite their remarkable achievements, a central question remains: What syntactic and semantic information do these contextualized word embeddings actually contain? To address this question, probing tasks are often employed. These tasks aim to measure how well linguistic information can be extracted from a pre-trained encoder. Researchers have shown that BERT can capture various linguistic features, including semantic roles, entity types, and semantic relationships (Turton et al., 2020; Tenney et al., 2019). Building on these insights, we wanted to investigate whether BERT also encodes knowledge about the category of animacy.

In this study, we particularly focus on the categorization of animacy and inanimacy in nouns, aiming to understand how these distinctions are represented and processed within BERT's embeddings. To achieve this goal, we employed probing tasks using a Multi-Layer Perceptron (MLP) classifier. The dataset was prepared using the POS dataset from Hugging Face, which we annotated as animate or inanimate using WordNet. The data were then tokenized using BERT, and embeddings were extracted. To explore this topic further, our investigation is guided by two primary research questions:
RQ1: How well does BERT capture the distinction between animate and inanimate nouns? This question examines the model's precision and effectiveness in recognizing and differentiating categories based on animacy within its contextual embeddings.
RQ2: How effective is the WordNet framework in annotating nouns as animate or inanimate? This question assesses the accuracy and reliability of WordNet as a tool for labeling and categorizing nouns.
In summary, for RQ1, our analysis suggests that BERT's ability to distinguish between animate and inanimate nouns is limited. Since accuracy could not be reliably used due to WordNet's high rate of misclassifications, we turned to cosine similarity as an alternative measure. The results show an average cosine similarity of 0.4980 for animate nouns and 0.4977 for inanimate nouns. These near-identical scores indicate that BERT struggles to ef-

fectively capture the distinction between these categories within its contextual embeddings. For RQ2, WordNet proved to be an unreliable tool for animacy annotation, with frequent misclassifications that limit its effectiveness in this context.

2 Related Work

Pre-trained models like BERT have been widely studied for their ability to capture and represent linguistic information across various layers of their architecture.

Tenney et al. (2019) demonstrated that BERT captures linguistic information in a hierarchical manner, with lower layers encoding syntactic features and higher layers handling more complex semantic tasks. Their work provides a foundation for probing specific linguistic phenomena within BERT's architecture. Building on this, we focus on the animacy distinction in nouns, exploring whether BERT's contextualized embeddings contain information about this semantic category.

Zheng et al. (2023) further investigated BERT's syntactic knowledge, showing that certain attention heads effectively encode specific syntactic relations. Their findings highlight BERT's strong performance in sentence parsing, which reinforces our focus on probing how BERT handles semantic distinctions like animacy.

Kauf et al. (2022) explored the semantic properties of large language models (LLMs), demonstrating that pre-trained LLMs can distinguish between possible and impossible events, showcasing strong performance at the syntactic level but less robustness at the semantic level. Their findings emphasize the importance of probing deeper semantic phenomena in models like BERT, which aligns with our goal of examining how well BERT recognizes animacy and inanimacy in nouns.

These research efforts underscore the relevance of probing BERT's representations of both syntactic and semantic information. In this paper, we apply probing techniques to investigate how well BERT's contextualized vectors capture the animacy distinction in nouns, contributing to the broader understanding of BERT's comprehension abilities in specific linguistic domains.

WordNet has long been a key resource for linguistic annotation, providing a structured semantic network of English words and their relationships (Miller, 1995). The study by Jahan et al. (2018) introduced a novel approach to animacy detection, moving beyond traditional word-level classification to focus on co-reference chains. Their method uniquely combines supervised machine learning with hand-crafted rules that leverage the hypernymy structure of WordNet. Inspired by this approach, we applied WordNet to annotate nouns in our study, specifically focusing on the animacy distinction.

3 Dataset

The dataset for this project is sourced from the Hugging Face platform, available at: Hugging Face POS Tagger Dataset. The original dataset contains a total of 8,194 sentences with 198,796 tokens annotated with Part-of-Speech (POS) tags. For our project, we focused on the following noun categories annotated in the dataset:

- **NN** (27,667 instances): Represents a Noun, singular or mass.
- **NNP** (19,374 instances): Denotes a Proper noun, singular.
- **NNS** (12,656 instances): Used for Noun, plural.
- **NNPS** (558 instances): Indicates Proper noun, plural.

Initially, the dataset contained a total of 60,255 nouns. See Figure 1.

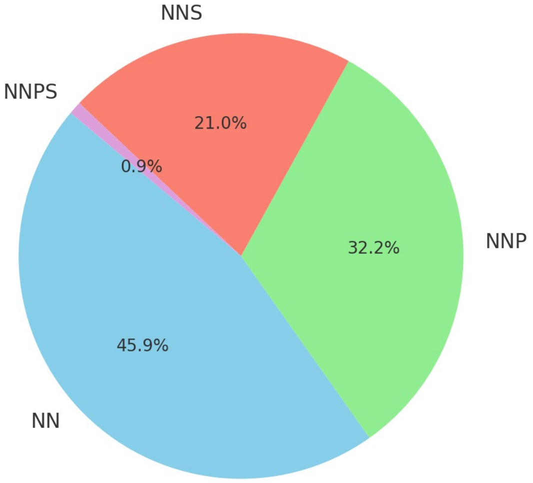


Figure 1: distribution of POS tags (NN, NNP, NNS, NNPS)

3.1 The formatted Dataset -

This format initially structured the data to include four key columns: "Sentences" contains the full sentences from the entire dataset."Sentence No" holds the corresponding sentence number for each sentence. For each word in a sentence, a designated "Word" column captures the individual words, and for each word, there is an associated "POS" column detailing the part-of-speech tags. This organization enables further annotation with animate and inanimate labels and facilitates the extraction of embeddings for each word within a sentence.

Data Annotation - To accurately extract the desired semantic information, the relationships of the nouns (noun classes) were analyzed using the WordNet ontology. This ontology was integrated into Python via the NLTK library to extract hypernyms for each corresponding word. Additionally, the nouns were categorized as either animate or inanimate based on their semantic properties.

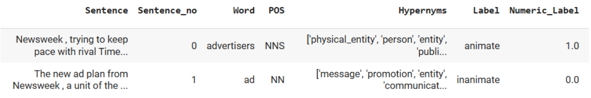
The annotation process involved structuring the hierarchical relations of words within WordNet, identifying higher-level categories, or hypernyms, to uncover broader semantic meanings. The focus was on the first sense of each word as a noun, providing a clear semantic context for further analysis.

A key aspect of this process was classifying the nouns as either ”animate” or ”inanimate,” based on the analysis of hypernyms. If the hypernyms indicated a relationship to the category of ”living things,” the noun was classified as animate (Label 1); otherwise, it was categorized as inanimate (Label 0). See Fig. 2.

This method generally helps to structure the hierarchical relationships between words and elucidate broader semantic meanings through hypernyms, but it also has its limitations. Specifically, the method relies on a straightforward categorization of ”living things” as animate; however, this simplification can lead to inaccuracies. For instance, although plants are technically living things, they are often not categorized as animate in linguistic contexts. Additionally, the polysemous nature of words can result in inconsistent categorizations: words like ”agents” may be classified as animate when referring to people, but as inanimate when referring to substances such as cleaning agents. Examples of misclassification by WordNet in our dataset include:

- Misclassified as inanimate: Words such as ”Agents,” ”Publisher,” and ”Founder,” which can denote humans in certain contexts.

- Misclassified as animate: Typically inanimate objects like ”Tree” and ”Flower” were erroneously categorized as animate.



Sentence	Sentence_no	Word	POS	Hypernyms	Label	Numeric_Label
Newsweek , trying to keep pace with rival Time...	0	advertisers	NNS	['physical_entity', 'person', 'entity', 'publi...	animate	1.0
The new ad plan from Newsweek , a unit of the ...	1	ad	NN	['message', 'promotion', 'entity', 'communicat...	inanimate	0.0

Figure 2: Annotated Data

Final Noun Labels During the data processing, several duplicates were removed. Additionally, many nouns that were not recognized by WordNet were also not annotated. As a result, a total of 45,427 nouns were labeled by WordNet, with 6,134 nouns (13.5%) classified as animate and 39,293 nouns (86.5%) classified as inanimate. See Figure 2 for a visual representation of this distribution.

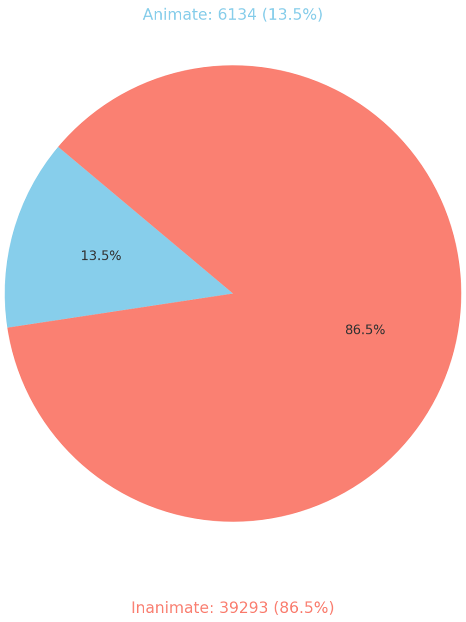


Figure 3: distribution of nouns as animate versus inanimate.

3.2 Extraction of Embeddings

To extract embeddings for each word, the sentences from the rows were used as input for the tokenizer. Since BERT uses the WordPiece approach, words that are not present in the tokenizer's vocabulary are split into subtokens. The sentences, provided in string format, were passed as input to the tokenizer. The BERT tokenizer, based on the WordPiece approach, decomposes words not found in its vocabulary into subtokens. Following the methodology used by the authors of the official BERT paper for word classification tasks, such as the NER task, this study employs the same approach. As this study also focuses on a word classification task, the first subtoken embedding was used as the representation for the entire word (Devlin et al., 2018). This approach represents a basic method, and future work could explore more sophisticated techniques to improve word representation, such as calculating the average of embeddings across all subtokens (Akdemir et al., 2020).

4 Transformer Models

In 2017, Google introduced an innovative approach to neural network architecture: the Transformer (Vaswani et al., 2017). This model achieved outstanding results in the field of Natural Language Processing (NLP) for sequence modeling, significantly surpassing the previously developed language models based on Recurrent Neural Networks (RNNs). This advancement marks a milestone in development and serves as a crucial foundation for the emergence of two major language models: Generative Pretrained Transformers (GPT) and BERT. Both models stand out due to the combination of the Transformer architecture with unsupervised learning approaches. This eliminated the need for training task-specific architectures from scratch. On the basis of GPT and BERT models, further Transformer-based models have been developed (Tunstall et al., 2023).

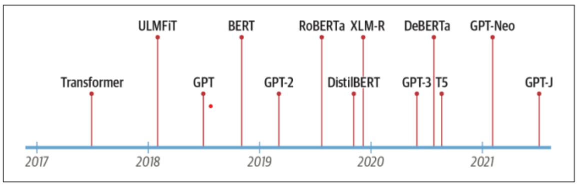


Figure 4: Chronological Evolution of Transformer Models (Tunstall et al.,2023).

4.1 BERT

In 2018, researchers from Google Research introduced a revolutionary model known as BERT, which stands for "Bidirectional Encoder Representations from Transformers." This model marked a significant turning point in the world of NLP. Following its introduction, BERT quickly gained recognition for its remarkable ability to achieve outstanding accuracy in a variety of NLP and Natural Language Understanding (NLU) tasks.

BERT's architecture consists of a stack of Transformer encoders with multiple self-attention heads, which weigh the importance of different words in a sentence, enabling a deep understanding of context. Unlike full Transformer architectures, which utilize both encoders and decoders, BERT employs only the encoder component of the Transformer. BERT is pre-trained on two unsupervised tasks: Masked Language Modeling (MLM), where it predicts randomly masked input tokens, and Next Sentence Prediction (NSP), where it predicts if two input sentences are adjacent.

After pre-training, BERT can be fine-tuned with minimal adjustments for specific tasks. Fine-tuning involves training the pre-trained model on a smaller, task-specific dataset with an additional task-specific layer on top. The entire model, including BERT and the task-specific layer, is fine-tuned jointly on the task-specific dataset, allowing BERT to adapt its representations to the nuances of the specific task. (Devlin et al., 2018)

4.2 BERT Input Representations

One of BERT's unique features is the use of WordPiece tokenization (Wu et al., 2016). This method segments tokens into their respective subtokens, which is particularly useful for handling unknown tokens that are not present in the vocabulary. For example, the word "playing" is tokenized into "play" and "##ing," as demonstrated in the figure 5 below. This process occurs before the tokens are encoded.

Neural networks (NN) fundamentally operate with numbers. Thus, all tokens are represented numerically. For instance, consider a vocabulary of 10 words where each word is assigned a unique index between 0 and 10. Each word can then be represented by its designated index.

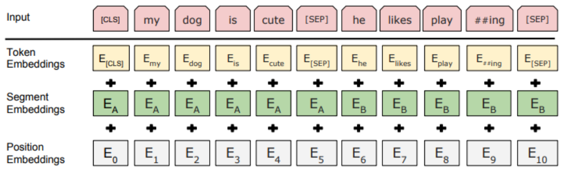


Figure 5: Bert Input Repräsentation (Devlin et al., 2018)

This Figure 5 illustrates how BERT combines token embeddings with segment and position embeddings to represent each word within its neural architecture, effectively capturing both the semantics and the structural information of the input text.

Following this tokenization principle, all our data were tokenized. In cases where more than one subtoken was present, only the main token was retained, and embeddings were extracted based on this. For further details on the extraction process, see Section 3.2 "Extraction of Embeddings."

5 Probing Architecture

For the implementation of our probing task, an MLP (Multilayer Perceptron) Classifier was employed, specifically trained with BERT's contextualized word representations to determine whether a noun is animate or inanimate. Figure 6.

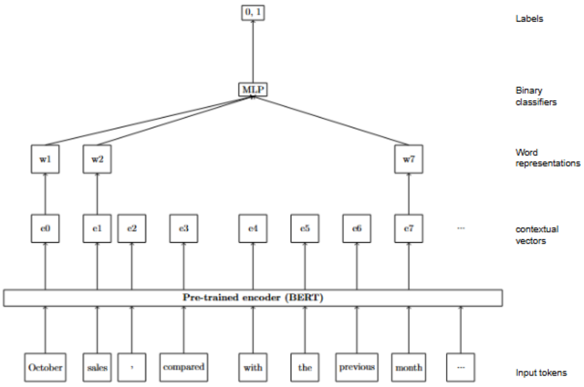


Figure 6: Probing model architecture, based on Tenney et al. (2019).

The process is structured as follows:

Input Tokens: The input consists of tokenized sentences processed by the pre-trained BERT model.

Contextual Vectors: BERT generates contextual embeddings for all tokens in a sentence. However, only the embeddings corresponding to the target nouns are extracted for further processing.

MLP Classifier: The extracted noun embeddings are passed into the MLP classifier, which predicts binary labels (0 or 1) to indicate whether the noun is animate or inanimate.

No Fine-Tuning: The pre-trained BERT encoder is used without fine-tuning. The MLP classifier operates solely on the frozen embeddings provided by BERT. (Tenney et al., 2019)

5.1 MLP Classifier

The MLP Classifier implemented in this project for the probing approach is designed as a feedforward neural network to tackle binary classification. The architecture is structured as follows:

Input Layer - The input layer is sized to match the dimensions of the BERT embeddings, ensuring that the model can effectively process the input data for classification.

Hidden Layers - The network includes two hidden layers, with 100 neurons in the first layer and 50 in the second. Both layers use the ReLU (Rectified Linear Unit) activation function, introducing non-linearity that enables the model to learn complex relationships between the input features and the target labels, which is crucial for distinguishing between animate and inanimate nouns.

Output Layer - The output layer consists of a single neuron with a sigmoid activation function, which outputs the probability that a noun is either animate or inanimate, making it appropriate for binary classification tasks.

Training Process - The model is trained using the Adam optimizer with a learning rate of 0.01 over 40 epochs. The loss function used is Binary Cross-Entropy (BCELoss), suitable for binary classification. Training data is converted into tensors for PyTorch compatibility, and the optimizer iteratively updates the model's weights to minimize loss.

Dataset Split - The dataset was split 80/20% for training and testing, resulting in 9,086 nouns in the test set. This ensures the model is evaluated on a sufficient portion of the data while preserving enough samples for effective training.

This MLP Probing Classifier is ideally suited for our specific classification task of distinguishing between animate and inanimate nouns. It leverages the capabilities of neural networks to effectively classify linguistic features based on the nuanced differences that the training with BERT embeddings provides (LeCun et al., 2015).

5.2 Performance Indicators

To evaluate the efficiency of the models, three main metrics are used:

Accuracy: This indicates how often the model makes correct predictions.

F1-Score: This metric combines precision (how many of the elements predicted as animate are actually animate) and recall (how many of the truly animate elements were identified by the model). It can be seen as a balance between precision and recall.

Confusion Matrix: This matrix helps to understand in which areas the model made errors. It shows how many animate and inanimate elements were correctly or incorrectly classified (Sokolova et al., 2016).

Additionally, **cosine similarity** with test data is used to evaluate the alignment between annotations of animate and inanimate objects, as well as nouns, and their corresponding word embeddings by measuring vector similarity. This value ranges from -1 (indicating exactly opposite meanings) to 1 (indicating exactly the same meanings).

6 Results

The probing classifier (MLP) was evaluated on a total of 9,086 nouns. Based on accuracy and F1-score, the model appears to perform very well. However, these results are not entirely realistic, as there are no gold-standard labeled data available, and the labels provided by WordNet contain many errors. The model's performance metrics are as follows:

Accuracy - The classifier achieved an overall accuracy of 0.977, indicating that the model correctly predicted the animacy of nouns in approximately 97.7% of cases.

F1-Score - An F1-Score of 0.923 indicates a balanced performance between precision and recall, effectively minimizing false positives and false negatives.

Confusion matrix provides insight into the performance of the MLP classifier when distinguishing between animate and inanimate nouns. It helps identify where the model succeeds and where it encounters difficulties. The matrix highlights the following key points:

- True Positives (TP): The model correctly predicted 1,108 words as animate, which are actually animate.

- True Negatives (TN): The model correctly predicted 7,774 words as inanimate, which are actually inanimate.

- False Positives (FP): The model incorrectly predicted 75 words as animate, but they are actually inanimate.

- False Negatives (FN): TThe model incorrectly predicted 129 words as inanimate, but they are actually animate. Figure 7.

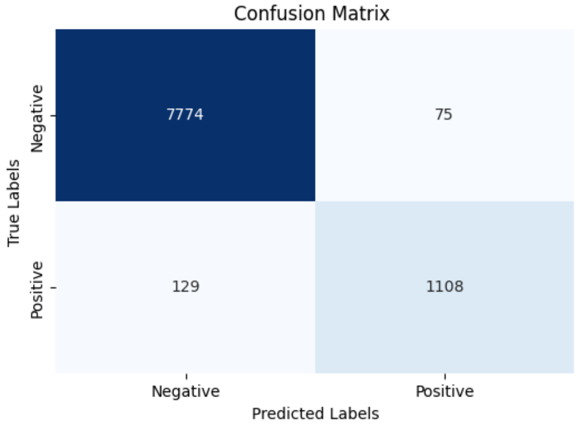


Figure 7: Confusion Matrix

6.1 Cosine Similarity Results

Without access to gold-standard annotated data, cosine similarity provides a more practical method for evaluating the alignment between annotations and nouns in terms of animacy. The cosine similarity results indicate that the model struggles to capture the relationship between the annotations and the nouns in terms of animacy. The detailed results are as follows:
- Animate Nouns The average cosine similarity was 0.498.
- Inanimate Nouns: The average cosine similarity was 0.497.

6.1.1 Confusion Matrix Based on Cosine Similarity

In this approach, cosine similarity was employed to evaluate the accuracy of noun classifications regarding their animacy ('animate') and inanimacy ('inanimate'). The analysis leveraged a threshold of 0.5 for cosine similarity. When values meet or exceed this threshold, a noun is considered correctly classified according to its actual label. Conversely, if the cosine similarity falls below 0.5, the prediction is flipped to the opposite of the actual label. A confusion matrix (Fig. 8) was generated to evaluate the model's performance, summarizing the following results:
- True Positives (961): The model accurately predicted 961 animate nouns as animate.
- True Negatives (6099): The model correctly identified 6099 inanimate nouns as inanimate.
- False Positives (276): The model mistakenly classified 276 inanimate nouns as animate.
- False Negatives (1750): The model erroneously labeled 1750 animate nouns as inanimate.

These findings reveal a significant number of misclassifications for animate nouns, pointing to potential areas for improving the model's ability to detect animate entities. This suggests the need to refine cosine similarity thresholds or integrate additional features to enhance category discrimination.

Unlike the prior evaluation using standard metrics with the probing MLP classifier, this approach utilizes cosine similarity, resulting in more accurate and realistic results. By offering a methodologically different measure, this analysis provides a more reliable assessment of classification accuracy.

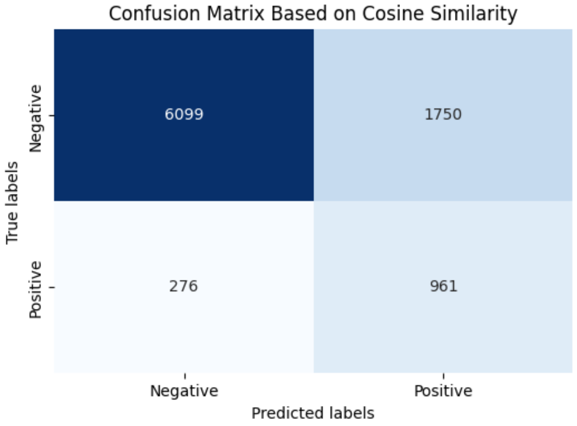


Figure 8: Confusion Matrix for Cosine Similarity

7 Conclusion

As previously mentioned, this study aimed to address two key research questions: (RQ1) How well does BERT capture the distinction between animate and inanimate nouns? and (RQ2) How effective is the WordNet framework in annotating nouns as animate or inanimate?

For RQ1, our findings based on cosine similarity scores suggest that BERT's ability to differentiate between animate and inanimate nouns is limited, with an average cosine similarity of 0.498 for animate nouns and 0.497 for inanimate nouns. The analysis in Section 6.2 shows that out of 9,086 tested nouns, 77.7% were correctly classified overall, with both animate and inanimate nouns exhibiting identical rates of 77.7% correctly identified and 22.3% misclassified.

For RQ2, our analysis revealed that WordNet is not a reliable tool for animacy annotation. The frequent misclassifications observed limit its effectiveness as a resource for labeling and categorizing nouns based on animacy. This lack of reliability further complicated the evaluation of BERT's performance.

In summary, while BERT shows promise in various NLP tasks, its ability to distinguish between animate and inanimate nouns within contextual embeddings remains weak. Furthermore, WordNet's limitations highlight the need for more accurate and consistent annotation frameworks in future research.

8 Limitations, Future Work

This study has identified several limitations that could be addressed in future research to improve both data annotation and model performance.

Manual Annotation - One key limitation is the reliance on automatic annotations, which introduced errors due to WordNet's misclassifications. Future research could improve reliability by using manually labeled gold-standard data, ensuring higher accuracy in evaluating model performance.

Automatic Annotation Limitations - While fully automated data annotation was used in this study, it has shown limitations in accuracy and reliability. Future work could explore semi-automated annotation methods, combining human oversight with automation to reduce errors. Additionally, alternative resources like ConceptNet or other frameworks could be experimented with, providing a more robust basis for data annotation than WordNet.

LLM Prompt Engineering - To enhance the annotation process, future research could explore different prompt engineering strategies with large language models (LLMs). This could lead to improved quality in data annotation by leveraging LLMs' ability to generate more contextually accurate prompts.

Subtoken Embeddings - In this study, only the embeddings of the first subtoken were used to represent the entire word, which is a simplification. Future research should consider calculating the average of embeddings across all subtokens to capture a more comprehensive word representation, potentially leading to better model performance.

In conclusion, addressing these limitations and exploring the proposed future directions could significantly enhance the accuracy and effectiveness of models like BERT in capturing complex semantic distinctions, such as animacy.

References

Matthew Peters , MarkNeumann, MohitIyyer, MattGardner, ChristopherClark, KentonLee, and LukeZettlemoyer. 2018a. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), volume 1, pages 2227–2237.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.0480

Mikolov, Tomas. ”Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781 (2013).

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

Tenney, I. ”BERT Rediscovers the Classical NLP Pipeline.” arXiv preprint arXiv:1905.05950 (2019).

Turton, Jacob, David Vinson, and Robert Elliott Smith. ”Deriving contextualised semantic features from bert (and other transformer model) embeddings.” arXiv preprint arXiv:2012.15353 (2020).

Zheng J, Liu Y. What does Chinese BERT learn about syntactic knowledge? PeerJ Comput Sci. 2023 Jul 26;9:e1478. doi: 10.7717/peerj-cs.1478. PMID: 37547407; PMCID: PMC10403162.

Kauf, C., Ivanova, A. A., Rambelli, G., Chersoni, E., She, J. S., Chowdhury, Z., ... & Lenci, A. (2022). Event knowledge in large language models: the gap between the impossible and the unlikely. arXiv preprint arXiv:2212.01488.

Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.

Jahan, Labiba, Geeticka Chauhan, and Mark A. Finlayson. ”A new approach to animacy detection.” Proceedings of the 27th International Conference on Computational Linguistics. 2018.

Vaswani, Ashish. ”Attention is all you need.” arXiv preprint arXiv:1706.03762 (2017).

Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. Natural language processing with transformers. ” O'Reilly Media, Inc.”, 2022.

Wu, Yonghui. ”Google's Neural Machine Translation System: Bridging the Gap between human and machine translation.” arXiv preprint arXiv:1609.08144 (2016).

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. ”Deep learning.” nature 521.7553 (2015): 436-444.

Akdemir, Arda, Tetsuo Shibuya, and Tunga Güngör. ”Subword contextual embeddings for languages with rich morphology.” 2020 19th IEEE International Conference on Machine Learning and

Applications (ICMLA). IEEE, 2020.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006, December). Beyond accuracy