Revisiting subword tokenization: A case study on affixal negation in large language models

Thinh Hung Truong[1] Yulia Otmakhova[1] Karin Verspoor[2,1]
Trevor Cohn[1,*] Timothy Baldwin[3,1]
[1]The University of Melbourne [2]RMIT University [3]MBZUAI
{hungthinht,yotmakhova}@student.unimelb.edu.au
karin.verspoor@rmit.edu.au trevor.cohn@unimelb.edu.au tb@ldwin.net

Abstract

In this work, we measure the impact of affixal negation on modern English large language models (LLMs). In affixal negation, the negated meaning is expressed through a negative morpheme, which is potentially challenging for LLMs as their tokenizers are often not morphologically plausible. We conduct extensive experiments using LLMs with different subword tokenization methods, which lead to several insights on the interaction between tokenization performance and negation sensitivity. Despite some interesting mismatches between tokenization accuracy and negation detection performance, we show that models can, on the whole, reliably recognize the meaning of affixal negation.

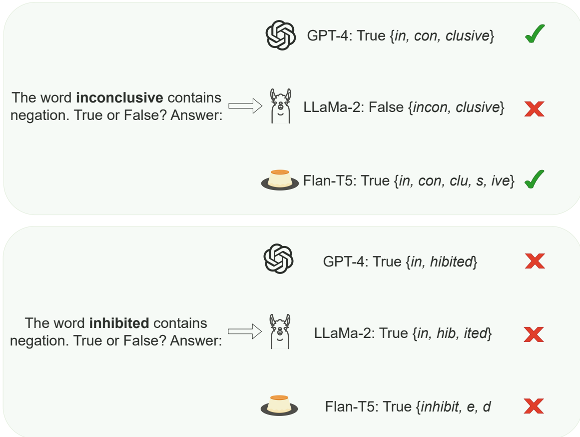


Figure 1: Example of our affixal negation prediction task, with the tokenization output for each model.

1 Introduction

Negation is central to language understanding but is not properly captured by modern NLP methods (Hossain et al., 2022; Truong et al., 2023, inter alia). While state-of-the-art large language models (LLMs) have improved negation-related capabilities, challenges remain, such as the ability to correctly determine the enclosed scope of negation, or when negation interacts with other linguistic constructions like quantifiers (She et al., 2023; Truong et al., 2023). Negations in common English NLP benchmarks are typically marked by separate negation cues such as *not* or *no*. However, in practice, negation can also be expressed through morphemes (or affixes) of words, i.e. by negative prefixes or suffixes such as in *uninteresting* or *effortless*.

While humans can identify affixal negation by leveraging morphological cues, NLP systems only rarely consider word-internal structure, beyond normalizing syntactic variation (Liu et al., 2012). Modern NLP methods such as language models employ subword tokenization, in which words are broken down into smaller units. This has an advantage of reducing vocabulary size, as well as learning shared representation between words with similar subwords. The intent to improve such representation by making tokenization methods more linguistically sound has driven the invention of several morphology segmentation methods, such as Morfessor (Grönroos et al., 2014). However, these have not been broadly adopted in modern LLMs as they do not scale well.

We hypothesize that current subword tokenization methods could lead to sub-optimal performance on language understanding tasks involving negation, because they do not correctly break words down morphologically. For instance, Table 1 demonstrates how different models employing different subword tokenization methods tokenize the word *anticlinal*. Another known challenge which could affect models is the high false positive rate in detecting affixal negations (Blanco and Moldovan, 2011), for example misinterpreting *de* in *deserve* as a negative affix, where in practice the *de* prefix derives from the Latin root *deservire* and should not be interpreted as negating *serve*.

In this work, we analyze the impact of affixal negations on transformer-based language models,

*Also at Google Research.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5082–5095
June 16-21, 2024 ©2024 Association for Computational Linguistics

Model	Type	Variant	Data source	Vocab. size	Output	NegMorph
BERT	BPE	WordPiece	books, wiki	30K	{*anti, clin, al* }	Correct
RoBERTa	BPE	Byte-level BPE	books, wiki	50K	{*antic, l, inal* }	Under-segmented
XLNet	ULM	SentencePiece	book, wiki, web text	32K	{*anti, clin, al* }	Correct
AlBERT	ULM	SentencePiece	book, wiki	32K	{*anti, clin, al* }	Correct
T5	BPE	SentencePiece	web text	32K	{*anti, clin, al* }	Correct
Llama-2	BPE	SentencePiece	web text, code, books, wiki, scientific publications	32K	{*ant, ic, l, inal* }	Over-segmented
GPT-2	BPE	Byte-level BPE	web text	50K	{*antic, l, inal* }	Under-segmented
GPT-4	BPE	Byte-level BPE	undisclosed	100K	{*antic, l, inal* }	Under-segmented

Table 1: Summary of different tokenizers used in our experiments. Output are tokenized version of the word *anticlinal* (model-specific special tokenization characters are removed for clarity purposes). All models are the base version unless otherwise specified.

where two main tokenization methods are employed, namely: byte-pair encoding (Gage, 1994; Sennrich et al., 2016) and unigram language model (Kudo, 2018). We consider three research questions:

RQ1: Are current subword tokenization methods able to preserve negative affixes? We analyzed the performance of various subword tokenization methods used in modern LMs. We find that most do not effectively produce the correct negative affixes.

RQ2: Are modern LMs aware of the presence of negation in affixal negations? We design a negation prediction task to probe models' awareness of affixal negation. We find that despite not performing well on the tokenization task, current LLMs can reliably infer the negated meaning of words with negative affixes. For this task, there is only a weak positive correlation between tokenizer and classifier performance.

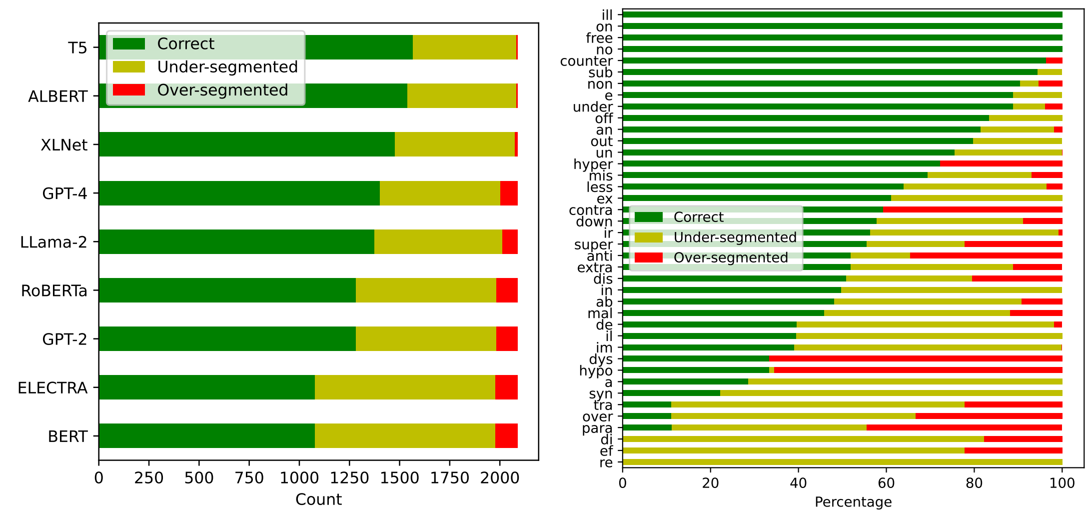
RQ3: What are the impacts of affixal negation on downstream tasks? As negation and sentiment are closely related, we measure the impact on a downstream sentiment analysis task by looking at samples containing affixal negations from common datasets. Results show that models perform well on those samples, implying that the impact of affixal negation is minimal. However, there exists a bias in predicting negative sentiment for affixal negations.

2 Related work

There are two popular ways of constructing the vocabulary for LMs using subword tokenization methods: byte-pair encoding ("BPE": Sennrich et al. (2016)) and unigram language model ("Unigram LM": Kudo (2018)). BPE starts from a base character set, then merges those characters based on bigram frequency to form subword units (bottom-up), whereas unigram language models start from a large subword vocabulary, which is then reduced based on a regularization method (top-down). There are multiple variants of BPE, differing in how the base vocabulary is represented and how the merging is done. WordPiece (Schuster and Nakajima, 2012) uses characters to represent the base vocabulary, then selects pairs that maximize the likelihood of training data, Byte-level BPE uses bytes instead of Unicode to represent the base vocabulary; the merging is done based on the frequency count of bigrams. In contrast, the unigram language model starts from a large base vocabulary and iteratively trims down tokens based on unigram LM perplexity until a target vocabulary size is reached.

Both methods assume that the input text uses spaces to separate words, which is not true for languages such as Chinese or Vietnamese. Therefore, a word segmentation step must be performed in advance. SentencePiece (Kudo and Richardson, 2018) was introduced to solve this problem by considering whitespace as part of words, essentially treating the whole input stream as the smallest unit to perform tokenization on. Then, either BPE or unigram LM can be applied to construct the vocabulary. Regardless of method, they purely rely on statistical information and thus are not expected to produce morphologically-aligned subword tokens.

There have been efforts to build linguistically-sound word tokenization methods, most notably Morfessor and its variants (Grönroos et al., 2014, 2020). Building morphology-aligned segmentation methods, especially in a multilingual setting, is an active line of research through recent SIGMOR-



(a) NegMorph by models (ALBERT and XLNet use ULM, while the rest employ BPE)

(b) NegMorph by affixes

Figure 2: Negative affix-preserving segmentation performance on the set of affixal negations (van Son et al., 2016).

PHON shared tasks (Batsuren et al., 2022). These methods outperform general tokenizers in producing morphologically-aligned tokens, but their benefit on downstream tasks is often negligible (Domingo et al., 2019; Saleva and Lignos, 2021). In this work, we examine if morphologically correct tokenization is important for LLMs to deal with negation.

BERT and its variants have been shown to be insensitive to negation (Kassner and Schütze, 2020; Ettinger, 2020), affecting many downstream NLP tasks such as sentiment analysis, NLI, or QA (Hossain et al., 2020, 2022; Ravichander et al., 2022; Truong et al., 2022). Compared to previous models, current LLMs have improved negation handling ability, but still struggle with some unconventional types of negation and linguistic constructions (Truong et al., 2023). Here, we investigate the treatment of affixal negation in modern LMs, with the intuition that subword tokenization methods that don't appropriately reflect this morphology will lead to misinterpretation of their semantics.

3 Experimental settings

We focus our analysis particularly on how affixal negations are represented in modern LLMs, designing probing tasks to test their awareness of negation, and the effect on downstream tasks. All code for the experiments is available at `https://github.com/joey234/affixal-negation`.

3.1 A lexicon of affixal negation

We use the lexicon created in van Son et al. (2016). The dataset contains a list of affixal negation and their non-negated counterparts (e.g. *unintended–intended*). For each affixal negation, the corresponding negative affix is also annotated. In total, there are 2089 affixal negations, and 2055 non-negated words which are antonyms of the negations. These numbers are not equal because one word can have multiple corresponding negated counterparts, e.g. *intrusive–{extrusive, unintrusive}*.

3.2 Tokenization methods

For each tokenizer type (along with their variants), we consider the most representative models that use them, based on their popularity. Although some models use the exact same tokenizer, it is worth investigating them as differences in training corpora can lead to differences in tokenization results.

BPE We consider models using different flavors of BPE. For WordPiece, we consider BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020); for Byte-level BPE, we consider RoBERTa (Liu et al., 2019), and GPT-family models including GPT-2 (Radford et al., 2019) and GPT-4 (OpenAI, 2023); and for SentencePiece, we examine Flan-T5 (Chung et al., 2022) and LLaMA-2 (Touvron et al., 2023).

Unigram LM Models using unigram LM tokenization methods considered in this work are always used in combination with SentencePiece: XL-

Net (Yang et al., 2019) and AlBERT (Lan et al., 2020).

3.3 Negative affix-preserving segmentation

We consider a segmentation of an affixal negation to be "affix preserving" (**Correct**) only if the negative affix matches with one of the produced tokens (e.g. *anticlimatic* → *anti, clima, tic*). Otherwise, it is either **Under-segmented** if the negative affix is a substring of one of the produced tokens (e.g. *anticlima, tic*), or **Over-segmented** (e.g. *ant, i, clima, tic*). Formally, given an affixal negation word w having the negative affix a, if w is tokenized into $T_k = \{t_i, t_{i+1}, ..., t_n\}$ under tokenizer k then we define $\text{NegMorph}_k(w)$ as follows:

$$\text{NegMorph}_k(w) = \begin{cases} \text{Correct} & \text{if } a \in T_k. \\ \text{Under-segmented} & \text{if } a \text{ is a substring of any } t_i \in T_k. \\ \text{Over-segmented} & \text{otherwise} \end{cases}$$

4 Findings

4.1 Current subword tokenization methods are not negative affix-preserving

As shown in Figure 2a, T5 has the best performance in producing negative affix-preserving tokens, while for the remaining, models employing the unigram LM method outperform those using BPE. This is in line with previous findings that the unigram LM produces subword units that align with morphology better than BPE (Bostrom and Durrett, 2020). Moreover, models that employ SentencePiece (T5, ALBERT, XLNet, LLaMA-2) outperform those that don't (BERT, RoBERTa, GPT-2). However, the best-performing models are only up to 75% correct relative to NegMorph, with considerable room for improvement. Most failed cases relate to under-segmentation.

An analysis of what types of negative affix are hard to tokenize is provided in Figure 2b, and their most frequent incorrect tokenizations are shown in Figure 3. Some common affixes that are incorrectly tokenized are *il* → *ill* (*illicit, illogical*), *ir* → *irre* (*irresolute, irreponsibly, irregular*), *a* → *as* (*asymmetric*), and *a* → *at* (*atypically*). Overall, we see that some affixes can be wrongly tokenized in a wide range of ways (represented by the large number of substacks), showing that current tokenization methods are inefficient. Overcoming this problem

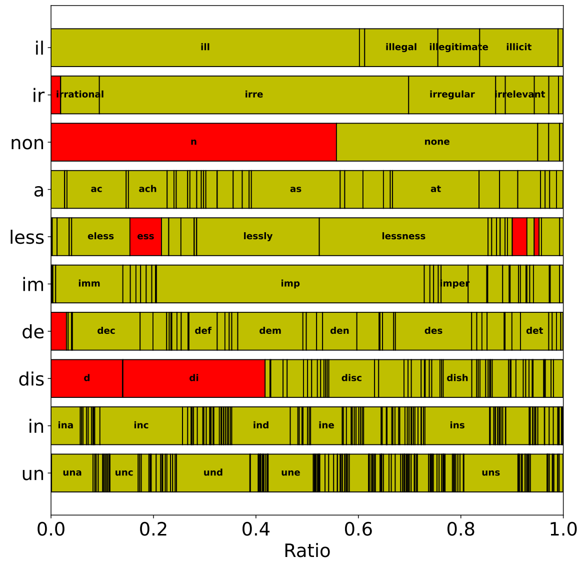


Figure 3: Top 10 most frequent affixes in the dataset and the distribution of tokens that they are wrongly tokenized into. Each substack denotes the percentage that the corresponding token is accounted for. Yellow bar denotes Under-segmented, while Red bar denotes Over-segmented.

could result in embeddings that better encapsulate word morphology.

4.2 Negative affixes signify negation, but word knowledge is essential

We design a binary classification task on the lexicon described in Section 3.1 to probe the ability of models to understand affixal negation, denoted *Affix*. First, for smaller models (<1B parameters), we conducted a fully fine-tuned setting with a 80/20 split and see that they achieve good results on the test set (>93% accuracy), showing that models can learn the patterns of negative prefixes and suffixes with enough supervision.

For larger models, we evaluate three state-of-the-art LLMs in a zero- and few-shot manner. The prompts are presented in Figures 5 and 6, respectively.

For the few-shot prompt, we provide explicit instructions to explain what negation means in this context, as well as two demonstrating samples, to avoid ambiguity (such as confusion with negative sentiment).

Results are summarized in Figure 7 (full numerical results, including in the fine-tuned setting, are in Table 4). Overall, we find that the performance on Neg (the subset containing only affixal negation) is much lower compared to its Non-neg counter-

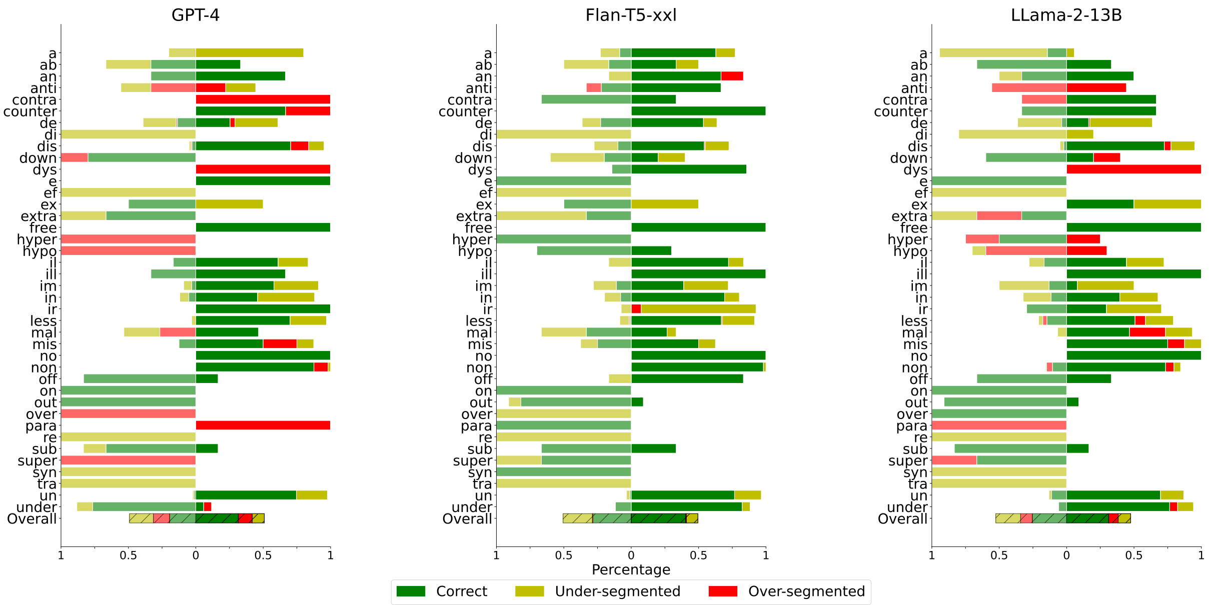


Figure 4: Ratio of correct/incorrect prediction on the Affix (fewshot) task, breakdown by affixes. The left greyed-out side of each subplot corresponds to wrong predictions.

Affix (zero-shot)

```
The word {word} contains negation. True or
False?
Answer:
```

Figure 5: Zero-shot prompt

part, where the best models achieve near-perfect performance.

For the zero-shot setting, surprisingly, Flan-T5 outperforms both LLaMA-2 and GPT-4, despite being the smallest in size. After adding more explicit instructions and examples (Affix (few-shot)), we observe large increases in performance for GPT-4 and LLaMA-2, and little to no difference for Flan-T5. For the non-negated subset, on the other hand, all models have near-perfect performance, with GPT-4 slightly outperforming Flan-T5. LLaMA-2 performance for this task is much lower than the other two.

We further break down the results based on affixes. Figure 4 illustrates the percentage of correct/incorrect prediction for each affix, divided by NegMorph categories. Compared to the relatively high results for Neg in Table 4, we have a clearer view on the actual performance of models. On average, we see that models made errors equally as likely for all affixes (as shown by the last Overall bar, where the percentages of incorrect and correct predictions are roughly 50%). From the figure,

Affix (few-shot)

```
A word contains negation if it has a negated
meaning, usually expressed through a negative
prefix (such as un, in) or suffix (such as
less).

The word decentralize contains negation. True
or False?
Answer: True
Explanation: decentralize is created by
prepending the root word centralize with the
negative prefix de.

The word deserve contains negation. True or
False?
Answer: False
Explanation: deserve just coincidentally starts
with de.

The word {word} contains negation. True or
False?
Answer:
```

Figure 6: Few-shot prompt

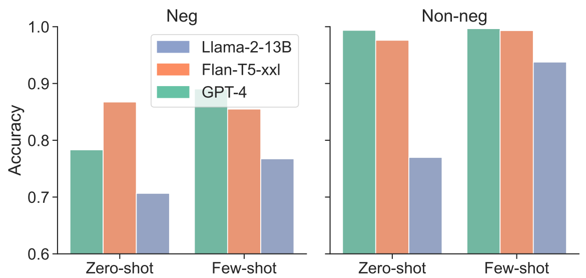


Figure 7: Zero- and Few-shot results on the affixal negation prediction task.

we can also observe that the correct/incorrect pre-

Error type	Example	Ratio
Reversal of action	*divest, diverge, detach*	0.213
Reversal of direction	*outdoors, descending, downstairs*	0.204
Insufficiency	*hypoglycemia, inferior, underpay*	0.132
Positive sentiment	*fearless, indispensable, unselfishly*	0.081
Wrongness	*infamy, malignant, misconstruction*	0.047
Rare words	*asyndetic, abactinal, syncategorem*	0.047
Noise (annotation errors)	*uncle, intense, increment*	0.106
Other (normal affixal negation)	*illicit, immortal, informality*	0.17

Table 2: Error analysis of the 235 errors made by GPT-4 on Affix (few-shot).

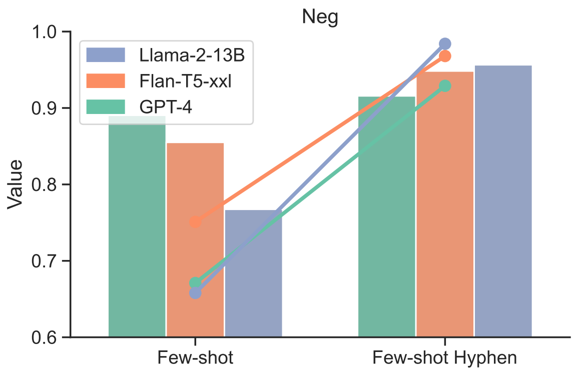


Figure 8: Results of Few-shot and Few-shot Hyphen on the affixal negation prediction task. Bars denote the accuracy on the prediction task , while Dots denote the Correct NegMorph scores for the segmentation task.

diction distribution is similar across models (especially between GPT-4 and Flan-T5), showing that they tend to make the same errors. In general, we see a larger portion of correct segmentation when models predict negation correctly, and more under-segmentation when models predict non-negation, while over-segmentation appears equally likely regardless of prediction. We also attempted to calculate the Pearson's coefficient between NegMorph and Accuracy on the Neg set but did not yield any statistically significant correlation.

Error analysis We inspect the errors made by GPT-4 in the Affix (few-shot) setting to perform a qualitative analysis. We adopt the classification introduced in Joshi (2012) and summarize in Table 2.

In total the model made 235 errors. Aside from the errors on normal affixal negation where the negative suffix can be replaced by *not* without changing the word's meaning (*uncritically, infinitely*), the bulk of the errors were caused by cases where the affix has more complex semantics. The most common source of errors is from affixes which show the reversal of action (e.g. *diverge, detach*), or the reversal of direction (e.g. *descending, downstairs*). In addition, a large proportion of errors come from affixes that negate reaching some normal or default state (e.g. *hypoglycemia, inferior, underpay*). Another interesting pattern is caused by words with positive sentiment (e.g. *fearless, incredibly, infallibility*) showing that models are confused between negation and negative sentiment, likely due to over-exposure to sentiment analysis data. In addition, some errors are attributed to words where the negative affix has an additional sentiment of "wrongness" (e.g. *malignant, misconstruction*). The remaining errors are attributed to rare words and noise in annotation.

Hyphenated words To make sure that the negative affix is not further broken down by tokenizers, we convert words into their "hyphenated" form (e.g. *unintended* $\rightarrow$ *un-intended*). From Figure 8, we see that this greatly increases the performance of different tokenizers on the NegMorph metric (by as much as 32%). Compared to the normal setting, the accuracy of all models also increases on the *Affix* task, suggesting a positive correlation between NegMorph and Accuracy. LLaMA-2 benefited the most from this setting, having the largest increases in both Accuracy and NegMorph.

Nonce words Nonce words are words that look and sound like real words, but are created for a single-purpose use and not recognized as words within a language (e.g. *roagly*). To measure the effect of negative affix on word semantics, we construct a list of "affixal nonce words" by prepending or appending negative affixes to a list of nonce words. We collect a list of adjective nonce words from Cremers (2022) . For affixes, we used the list of 40 negative affixes provided in van Son et al. (2016) and collected 40 non-negative affixes (e.g. *auto-, bi-, -ism, -ful*).[1] For each nonce word, we prepend (or append) the affixes to form an "affixal nonce word". In total, the set consists of 11 nonce words $\times$ 80 affixes = 880 samples, evenly distributed between negated (e.g. *dis-roagly*) and non-negated (e.g. *auto-roagly*). We adopt the Affix

[1]We collected the affixes from https://litinfocus.com/120-root-words-prefixes-and-suffixes-pdf-list/

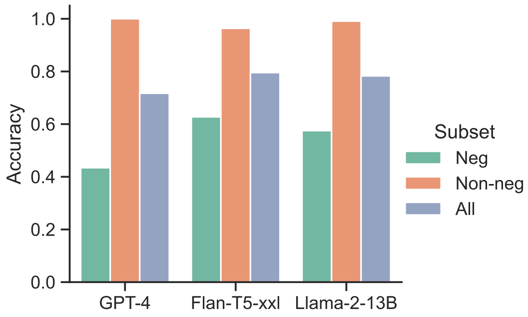


Figure 9: Accuracy on the affixal nonce words prediction task.

(few-shot) prompt and add an instruction to prevent models from refusing to answer the questions because of invalid words (full prompt in Appendix B). Similarly, we also report the results of two subsets of negative affixes (Neg) and non-negative affixes (Non-neg) in Figure 9. For the Neg set, we find that the performance of all models is relatively low, despite them being able to correctly tokenize the negative affixes. Whereas for the Non-neg set, performance is near-perfect for all models, similar to the previous Affix-Hyphen task. Looking at the results, however, we found that most errors made by the models are when the negative affixes are ambiguous, i.e. their meaning depends on which words they are attached to (e.g. *a-, di-, ef-, para-, re-*). This reveals an important insight that whether something is considered to be a negation should be judged with context (which is parametric knowledge about words in this case).

Non-negated words with tokens homonymous with negative affixes To explore the false positive problem i.e., words coincidentally contain negative affixes, we collect words from the Non-neg subset and WordNet (Miller, 1995) which do not have negated meaning, but have a negative prefix/suffix as the first/last subword token. We tokenize WordNet using the T5 tokenizer and select all words that start/end with a negative prefix/suffix, then subtract all words in the list of affixal negations. We manually go through the extracted list again to remove errors, resulting in a set of 330 words.[2] Following the same affixal negation prediction task, we find that Flan-T5 has very good performance (0.958 accuracy), showing that it can synthesize information from all subword tokens instead of only relying on the negative affixes. Most errors come from the *uni-* cases, where the model tokenizes them as *un-* (e.g. *unidirectional, univalent*).

4.3 Impact on downstream tasks

One main drawback of our probing task is that the words lack context. Negation is a context-dependant concept, that is, what is considered negation can differ depending on the context of use. Investigating the impact of affixal negation in the context of downstream tasks is thus an essential component of this work.

4.3.1 Sentiment analysis

Previous work has shown that negation is a strong indicator of negative sentiment (Wiegand et al., 2010). Furthermore, the fact that sentiment analysis is part of many NLP benchmarks could create a bias in models, leading to negation being conflated with negative sentiment, which is not always the case. For instance, the word *incredible* is constructed by prepending the root word *credible* with the negative affix *in-*, meaning "not credible" but used to express a positive meaning. This inspired us to extend our analysis to a downstream sentiment analysis task. We evaluate the few-shot performance of LLMs in two settings of word- and sentence-level sentiment analysis (full prompts in Appendix C).

Word-level sentiment We first use SentiWordNet 3.0 (Baccianella et al., 2010) to automatically assign a sentiment label for the lexicon of affixal negation described in Section 3.1. After that, two graduate researchers went over the list to determine the final labels (positive, negative, or neutral). In general, we find that GPT-4 outperforms Flan-T5 and LLaMA-2 on this word-level task. As seen in Figure 10, all models have almost perfect performance at predicting negative words, but struggle with the other two classes. In particular, we find Flan-T5 and LLaMA-2 overpredict Negative for Neutral words, while GPT-4 often mistakes Positive for Neutral.

Sentence-level sentiment For this task, we look at common sentence-level sentiment analysis datasets including SST-2 (Socher et al., 2013), and Rotten Tomatoes (RT) (Pang and Lee, 2005). One drawback of this evaluation is that samples tend to contain many sentiment signals, making it hard to gauge the effect of affixal negations.

[2] We didn't consider other models as this list of words would be different between models.

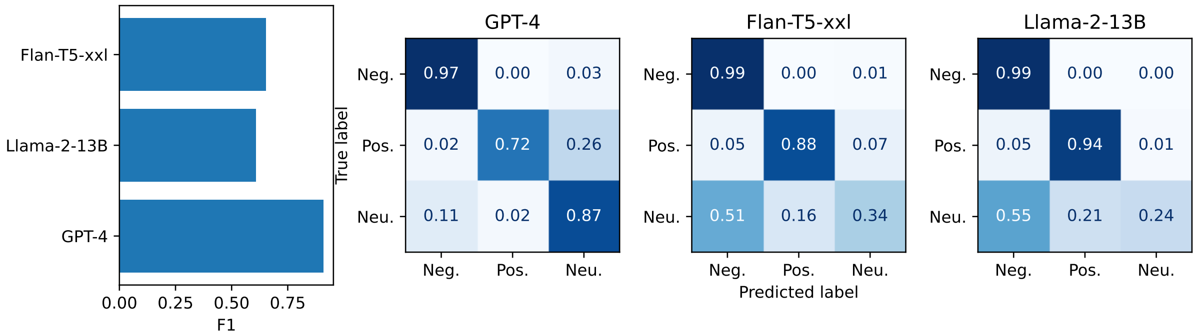


Figure 10: Performance on word-level sentiment task

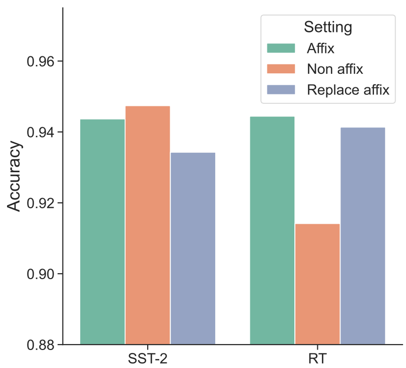


Figure 11: Accuracy on sentence-level sentiment analysis task. Results are averaged across 3 models.

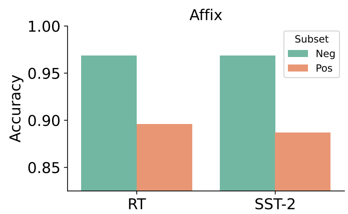


Figure 12: Accuracy of Neg/Pos class of the Affix set. Results are averaged across 3 models.

We consider 3 settings: (1) Affix = only samples containing affixal negation; (2) Non affix = only samples without affixal negation; and (3) Replace affix = similar to Affix, but we replace all instances of affixal negations with equivalent syntactic negations, i.e. *not* + word (*uninteresting* → *not interesting*). We present the results in Figure 11. While it is true that replacing negative affixes with *not* does not always result in a direct paraphrase, we argue that the change in meaning is minimal, and that samples will likely preserve the sentiment. Furthermore, we mostly find adjectives in the datasets rather than nouns, which ensures that the sentences are grammatically correct after replacement. Overall, we can conclude that affixal negation is a strong signal to guide model prediction. We observe good performance for Affix in both datasets, where the accuracy are comparable to Non Affix in SST-2 and higher in RT. Attempting to replace affixal negations slightly decreases the performance of models in both datasets. This suggests that affixal negation is actually a stronger sentiment cue compared to syntactic negation. We further report class-wise performance of the Affix set in Figure 12. Accuracy on samples with Negative sentiment is higher than Positive, once again showing that affixal negation is a strong cue for predicting negative sentiment.

5 A look into token attribution

We perform an interpretation analysis to attain insights into what drives model predictions. For this analysis, we use the Flan-T5-xxl model, as we could not obtain probabilities (logprobs) from GPT-4. We calculate the attribution for each token corresponding to the predictions using the Integrated Gradient method (Sundararajan et al., 2017), with probability as the scoring function, implemented in *Inseq* (Sarti et al., 2023). Overall, we observe high attribution scores from relevant tokens, such as the subword tokens of the target words, showing that models know where to pay attention to when performing inference.

Negative affixes have flipping sentiment effect In Section 4.3.1, we see that models tend to overpredict negative sentiment on the list of affixal negations. Through the saliency heatmap in Figure 13, we can see high token attributions for the negative affixes that change the sentiment of the root words (either positive or neutral) into negative. This is in line with previous findings that negation flips the polarity of sentiment (Tigges et al., 2023). This

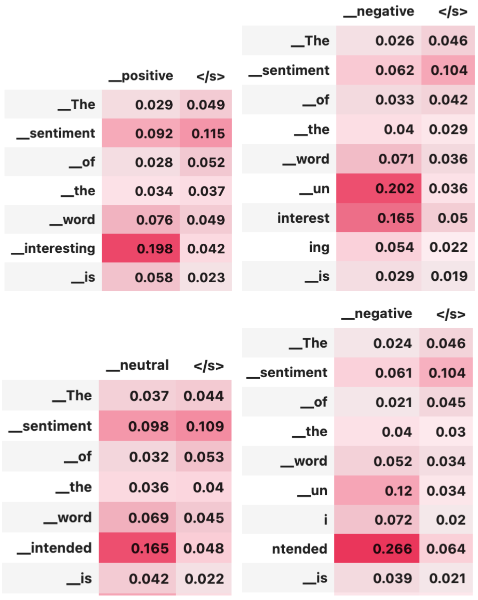


Figure 13: Token attribution of selected samples on the word-level sentiment prediction task. Only parts of the prompts are shown for clarity purposes.

effect could be the main cause for the low performance on the Neutral class observed in our word-level sentiment analysis task (Section 4.3.1). When applied to the negation prediction task, however, we did not observe a similar effect and did not see any clear pattern for token attribution.

Correct tokenization is not essential for negation awareness Through many experiments, we have shown that overall, correct tokenization leads to better awareness of models to the presence of negation. This effect, however is not significant. By comparing token attributions between 3 cases of NegMorph (Figure 14), we saw that models are able to combine information from relevant subword tokens corresponding to a word to make the correct inference.

6 Conclusion

In this work, we conducted an in-depth analysis into how well modern LLMs handle affixal negation, a type of negation where morphology is essential to understanding word semantics. We have shown that there is significant room for improvement in current tokenization methods in terms of producing negative affix-preserving tokens. Despite that, the effect of morphologically incorrect tokenization on the ability of models to understand word meaning in downstream tasks, including sentiment analysis, is minimal. Regardless, designing better subword tokenization methods may have many immediate benefits such as reducing vocabulary size, learning better word representations, and improving model interpretability.

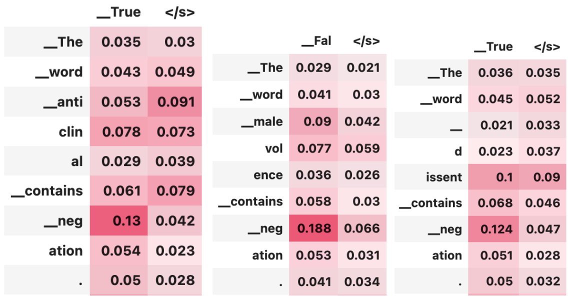


Figure 14: Token attribution of selected sample samples on the negation prediction task. The three subplots correspond to Correct, Under-segmented, and Over-segmented case respectively. Only parts of the prompts are shown for clarity purposes.

7 Limitations

Prompting As this work involves experiments with LLMs, there is always a possibility that the prompts we used are not optimal (and also, the problem of reproducibility). We attempted to reuse prompt templates from existing work where possible, and strove to design prompts that are intuitive and specific otherwise.

Multilinguality Morphology is a language-dependent problem. We recognize that the lack of investigation in other languages other than English is a drawback of this work.

Broader impact Given that our focus is on presenting and analysing the problem of poor treatment of affixal negation in LLMs, we did not propose any immediate solutions to improve the status quo. The finding on the impact on downstream tasks could be limited by the lack of samples (both in size and meaningful patterns) in the test data.

Acknowledgements

We would like to thank Khuyagbaatar Batsuren for the talk on morphological segmentation at The University of Melbourne that inspired this work, and Daniel Beck and Simon Šuster for discussions that helped refine our experiments. We also thank the anonymous reviewers and area chairs of ARR December cycle and NAACL for an engaging and constructive review period. This research was conducted by the Australian Research Council

Training Centre in Cognitive Computing for Medical Technologies (project number ICI70200030) and funded by the Australian Government. This research was undertaken using the LIEF HPC-GPGPU Facility hosted at The University of Melbourne, established with the assistance of LIEF Grant LE170100200.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Eduardo Blanco and Dan Moldovan. 2011. Some issues on detecting negation from text. In *Twenty-Fourth International FLAIRS Conference*.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv preprint*, abs/2210.11416.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations (ICLR 2020)*.

Alexandre Cremers. 2022. Interpreting gradable adjectives: rational reasoning or simple heuristics? *Empirical Issues in Syntax and Semantics*, 14:31–61.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Miguel Domingo, Mercedes García-Martínez, Alexandre Helle, Francisco Casacuberta, and Manuel Herranz. 2019. How much does tokenization affect neural machine translation? In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 545–554. Springer.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2020. Morfessor EM+Prune: Improved subword segmentation with expectation maximization and pruning. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3944–3953, Marseille, France. European Language Resources Association.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Md Mosharaf Hossain, Dhivya Chinnappa, and Eduardo Blanco. 2022. An analysis of negation in natural language understanding corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 716–723, Dublin, Ireland. Association for Computational Linguistics.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Shrikant Joshi. 2012. Affixal negation–direct, indirect and their subtypes 1. *Syntaxe & sémantique*, (1):49–63.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models:

Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations (ICLR 2020)*.

Haibin Liu, Tom Christiansen, William A Baumgartner, and Karin Verspoor. 2012. Biolemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of Biomedical Semantics*, 3:3.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv preprint*, abs/1907.11692.

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

OpenAI. 2023. GPT-4 technical report.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog.

Abhilasha Ravichander, Matt Gardner, and Ana Marasović. 2022. CONDAQA: A contrastive reading comprehension dataset for reasoning about negation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8729–8755, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jonne Saleva and Constantine Lignos. 2021. The effectiveness of morphology-aware segmentation in low-resource neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 164–174, Online. Association for Computational Linguistics.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, and Oskar van der Wal. 2023. Inseq: An interpretability toolkit for sequence generation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 421–435, Toronto, Canada. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *International Conference on Acoustics, Speech and Signal Processing*, pages 5149–5152.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jingyuan S. She, Christopher Potts, Samuel R. Bowman, and Atticus Geiger. 2023. ScoNe: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1803–1821, Toronto, Canada. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 101–114, Toronto, Canada. Association for Computational Linguistics.

Thinh Hung Truong, Yulia Otmakhova, Timothy Baldwin, Trevor Cohn, Jey Han Lau, and Karin Verspoor. 2022. Not another negation benchmark: The NaN-NLI test suite for sub-clausal negation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 883–894, Online only. Association for Computational Linguistics.

Chantal van Son, Emiel van Miltenburg, and Roser Morante. 2016. Building a dictionary of affixal negations. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics (ExProM)*, pages 49–56, Osaka, Japan. The COLING 2016 Organizing Committee.

Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. 2010. A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 60–68, Uppsala, Sweden. University of Antwerp.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

A Model endpoints

- GPT-4: We accessed GPT-4 through the offical API with the name `gpt-4`. Note that this is different from the GPT-4 turbo model with the name `gpt-4-1106-preview`.

- LLaMA-2-13B: We used the official instruction fine-tuned LLaMA-2-13B available on the HuggingFace hub with the name: `meta-llama/LLaMA-2-13b-chat-hf`.

- Flan-T5-xxl: We used the official xxl version (11.3B) of the Flan-T5 model available on the HuggingFace hub with the name: `google/flan-t5-xxl`.

B Details of Affixal Nonce word prediction task

List of nonce words *roagly, vibble, drok, scrop, plard, hif, tepable, plawic, bluth, sprat, flurf*

List of non-negative affixes Prefix: *ambi-, aqu-, ast-, aud-, auto-, bi-, bio-, cent-, circum-, co-, cred-, cycl-, dec-, dia-, equ-, geo-, grad-, hydro-, inter-, medi-, mega-, min-, micro-, pan-, semi-, tele-, uni-, tri-*. Suffix: *-able, -al, -ance, -ful, -ian, -ic, -tic, -ile, -ism, -ist, -junct, -ly*

Nonce

```
A nonce word is a word occurring, invented,
or used just for a particular occasion, or a
word with a special meaning used for a special
occasion. Infer whether the given nonce word
contains negation or not.

A word contains negation if it has a negated
meaning, usually expressed through a negative
prefix (such as un, in) or suffix (such as
less).

The word decentralize contains negation. True
or False?
Answer: True
Explanation: decentralize is created by
prepending the root word centralize with the
negative prefix de.

The word deserve contains negation. True or
False?
Answer: False
Explanation: deserve just coincidentally starts
with de.

The word {word} contains negation. True or
False?
Answer:
```

C Prompts for sentiment analysis

Word-level sentiment

```
{Few-shot samples}

The sentiment of the word {word} is positive,
negative, or neutral.

Answer:
```

Sentence-level sentiment

```
{Few-shot samples}

{sentence}
Question: Is this sentence positive or
negative?

Answer:
```

D Full results

Model	Neg. Nonce	Non-neg. Nonce	All
GPT-4	0.434	1	0.717
LLaMA-2-13B	0.575	0.991	0.783
Flan-T5-xxl	0.627	0.964	0.795

Table 3: Affixal nonce word prediction task

Model	Accuracy Neg	Non-neg	All	NegMorph Correct
Affix (fine-tuned)				
BERT	0.940	0.959	0.949	0.516
RoBERTa	0.931	0.964	0.947	0.614
AlBERT	0.933	0.956	0.945	0.737
XLNet	0.950	0.915	0.932	0.707
GPT-2	0.928	0.949	0.938	0.614
Affix (zero-shot)				
GPT-4	0.783	0.994	0.888	0.671
LLaMA-2-13B	0.707	0.770	0.738	0.658
Flan-T5-xxl	0.867	0.976	0.921	0.751
Affix (fewshot)				
GPT-4	0.890 (+0.107)	0.997 (+0.003)	0.943 (+0.055)	0.670
LLaMA-2-13B	0.767 (+0.060)	0.938 (+0.168)	0.852 (+0.114)	0.658
Flan-T5-xxl	0.855 (-0.012)	0.993 (+0.017)	0.924 (+0.003)	0.750
Affix (fewshot)-Hyphen				
GPT-4	0.916 (+0.133)	-	-	0.929 (+0.258)
LLaMA-2-13B	0.956 (+0.249)	-	-	0.984 (+0.326)
Flan-T5-xxl	0.948 (+0.081)	-	-	0.968 (+0.217)

Table 4: Results of our affixal negation prediction task. (+/- denote the change compared to the *Affix (zero-shot)* setting