How Much Semantic Information is Available in Large Language Model Tokens?

David A. Haslett
The Hong Kong University
of Science and Technology,
Hong Kong, China
haslett@ust.hk

Zhenguang G. Cai
The Chinese University of
Hong Kong, Hong Kong, China
zhenguangcai@cuhk.edu.hk

Abstract

Large language models segment many words into multiple tokens, and companies that make those models claim that meaningful subword tokens are essential. To investigate whether subword tokens bear meaning, we segmented tens of thousands of words from each of 41 languages according to three generations of GPT tokenizers. We found that words sharing tokens are more semantically similar than expected by chance or expected from length alone, that tokens capture morphological information even when they don't look like morphemes, and that tokens capture more information than is explained by morphology. In languages that use a script other than the Latin alphabet, GPT-4 tokens are uninformative, but GPT-4o has improved this situation. These results suggest that comparing tokens to morphemes overlooks the wider variety of semantic information available in word form and that standard tokenization methods successfully capture much of that information.

1 Introduction

Subword tokens in large language models (LLMs) are often compared to morphemes (i.e., productive constituents such as *pre* and *school*). In a popular tokenization resource (github.com/openai/tiktoken), OpenAI claims that LLMs ''will often split *encoding* into tokens like *encod* and *ing* (instead of e.g. *enc* and *oding*)'', and in another resource (huggingface.co/transformers), HuggingFace is explicit that ''rare words should be decomposed into meaningful subwords''. However, tokenization is typically determined by the frequency of strings of characters, not by semantics (for a review, see Mielke et al., 2021), so tokens don't reliably correspond to morphemes (Bostrom and Durrett, 2020; Church, 2020). Consider how three generations of GPT tokenizers segment *enc+odings*, *enc+od+ification*, and *enc+od+ified*, contra OpenAI's own example.

Perhaps LLMs treat tokens more like the letters of a very large alphabet, memorizing how to spell words token by token. Memorization can't account for unfamiliar words, though, and an important function of subword tokenization is to help LLMs represent rare words (Sennrich et al., 2015; Wu et al., 2016). This problem is compounded in low-resource languages, which, on average, make up less of an LLM's training data and contain more multi-token words (e.g., Petrov et al., 2024). Accurately representing rare words requires extracting information from word form, so in this study, **we investigate how reliably tokens imply word meanings and whether this varies across languages**. We build on previous research that compares tokens to morphemes (e.g., Hofmann et al., 2021, but we take a broader view of the semantic information available in word form, which includes etymological relationships and other patterns (e.g., *purse* and *bursary* or *glow* and *glisten*; for reviews, see Dingemanse et al., 2015, and Haslett and Cai, 2024).

1.1 Meaning in Large Language Model Tokens

Language is productive, so LLMs, like people, regularly encounter unfamiliar words. Roughly half of the words in any given dataset occur only once in that dataset (e.g., Baayen, 2012), and if anything, the proportion of single-use words increases in very large datasets (Fan, 2010). Jargon, neologisms, and typos spring eternal. Context plays an important role in representing the meanings of unfamiliar words, of course, but to reduce a word to its context is to extract no new information from that word, and the meaning implied by only a few contexts may not be representative,

Transactions of the Association for Computational Linguistics, vol. 13, pp. 408–423, 2025. https://doi.org/10.1162/tacl_a_00747
Action Editor: Nathan Schneider. Submission batch: 06/2024; Revision batch: 10/2024; Published 4/2025.

so computational models struggle with rare words (e.g., Sahlgren and Lenci, 2016; Schick and Schütze, 2020).

People interpret rare words, in part, by decomposing them to morphemes and by associating them with similar-sounding words (e.g., Alegre and Gordon, 1999; Haslett and Cai, 2023). Computational models such as FastText have adopted a similar approach, combining the representations of subword strings (Bojanowski et al., 2017; cf. Baayen et al., 2019; Marelli et al., 2015). For example, *encodings* combines *encod*, *ncodi*, *codin*, etc. Unlike FastText, LLMs segment words into non-overlapping, non-decomposable constituents, which may distort the information available in word form. GPT models use byte-pair encoding for tokenization, which segments text into common strings of bytes (Gage, 1994), and those LLMs learn patterns in tokens, without access to constituent bytes, so the information available in *cod* and *ing*, for instance, is obscured by the tokens *enc* and *odings* (though see Kaushal and Mahowald, 2022, for evidence that LLMs can infer the identify of some sub-token constituents).

Segmenting words into morphemes sometimes improves LLM performance, consistent with LLMs extracting information from meaningful constituents (Batsuren et al., 2024; Hofmann et al., 2022; Jabbar, 2023; Mager et al., 2022), but in other cases, the impact of morpheme-like tokens is negligible or the results are mixed (Arnett et al., 2024; Gutierrez et al., 2023; Park, 2020; Toraman et al., 2023). One explanation for this discrepancy is that when tokens diverge from morphemes, they might still convey valuable information. Notice how *enc+odings*, *enc+od+ification*, and *enc+od+ified* share the *enc* token. Similarly, Chinese words that share tokens often share submorphemic constituents called semantic radicals. As shown in Table 1, the Chinese characters for ‘‘suffer’’ and ‘‘dig’’, 挨 and 挖, contain the same semantic radical (扌, ‘‘hand’’, which commonly occurs in verbs), and they also begin with the same GPT-4 token. There may be method in the madness of frequency-based tokenization, allowing LLMs to glean meaning from seemingly arbitrary strings of bytes.

Language	Word 1	Word 2	Shared token
English	_clamped	_clapped	_cl
German	_innere (‘‘inner’’)	_innen (‘‘within’’)	_inn
Malay	_riang (‘‘carefree’’)	_ladang (‘‘farm’’)	ang
Chinese	挨 (‘‘suffer’’)	挖 (‘‘dig’’)	□ (e6, 8c)
Arabic	ضَاع_ (‘‘lost’’)	ذرة_ (‘‘corn’’)	_□ (20, d8)

Table 1: Pairs of words that share GPT-4 tokens in five languages. Underscores indicate word-initial whitespace. Chinese words are not separated by whitespace. Arabic is written from right to left, but the tokenizers incorporate left-leading whitespace into the rightmost (word-initial) byte. Arabic and Chinese characters comprise two and three bytes, respectively, so when tokens representing those scripts comprise fewer than two or three bytes, they lack surface-level representations and are instead represented with replacement characters (here, empty squares). Under replacement characters, pairs of hexadecimal digits identify bytes. The Chinese characters begin with the same two bytes (e6, 8c), and they contain the same semantic radical (扌, ‘‘hand’’). The Arabic words do not share any characters, but they do begin with the same byte (d8), along with word-initial whitespace (20). Of the 36,337 Arabic words in our sample, 63% begin with the d8 byte, which illustrates how unlikely single-byte tokens are to reliably imply word meaning in Arabic.

Whether tokens bear meaning is complicated by the fact that an LLM represents all languages with a single set of tokens. For example, GPT-2, GPT-4, and GPT-4o each segment the French words for ‘‘duck’’ and ‘‘rabbit’’ into *can+ard* and *lap+in*, which contain the same tokens as the English words for ‘‘can’’ and ‘‘lap’’. The training data for these LLMs is proprietary, but the skew towards English is apparent in their vocabularies. For example, the long words *telecommunications*, *interdisciplinary*, and *disproportionately* are each a single GPT-4 token. Petrov et al. (2024) counted GPT-4 tokens per word meaning and found that the language closest to parity with English in their sample, Portuguese, uses about 1.5 times the tokens that English does, while Shan, spoken in Myanmar, uses 15 times more tokens. The cruel irony of frequency-based tokenization is that words from lower resource languages are more likely to be segmented into letter-like tokens, yet LLMs have fewer opportunities to memorize how to spell out those words. If LLM performance depends on reliable representations of word meanings, then LLMs are likely confer greater professional or educational advantages on speakers of high-resource languages, which would contravene the mission statement of OpenAI: ‘‘to ensure that artificial general intelligence benefits all of humanity.’’

OpenAI has taken steps to address disparities in tokenization. GPT-4o uses fewer tokens than its predecessor, GPT-4, to represent scripts other than the Latin alphabet (OpenAI, 2024). For example, GPT-4 segments the three-letter Arabic word for ‘‘corn’’, ذرة, into four tokens, whereas GPT-4o segments it into two tokens. This makes it more plausible that GPT-4o represents word meanings in lower resource languages by decomposing multi-token words to meaningful constituents, though whether those tokens reliably imply word meanings is an open question.

1.2 The Present Study

We compared three generations of GPT tokenizers: GPT-2 (50,257 tokens, with the same vocabulary as GPT-3; Radford et al., 2019), GPT-4 (100,256 tokens; OpenAI, 2023), and GPT-4o (199,998 tokens; OpenAI, 2024). Each uses byte-pair encoding, which segments text into frequent strings of bytes. (GPT-1 uses a different tokenization method, spaCy.) GPT-2 is trained on predominantly English data, GPT-4 is proficient in a variety of languages, and GPT-4o more efficiently tokenizes languages that use a script other than the Latin alphabet. (In the online supplement, we compare these GPT tokenizers to the Multilingual BERT tokenizer, which was trained on 104 languages and uses WordPiece; Devlin, 2018.)

To investigate how reliably tokens imply meanings, we measured the semantic similarity of pairs of words that share tokens (e.g., *baton* and *bathe* share the GPT-4 token *bat*), compared to the similarity of pairs that share strings matched on length with tokens (e.g., *baton* and *glutton* share the three-letter string *ton*) and compared to randomly paired words (e.g., *baton* and *dog*). As explained below, we relied predominantly on distributional semantic models (e.g., word2vec; Mikolov et al., 2013), which in effect aggregate the linguistic contexts where a word occurs to represent its meaning, based on the assumption that words which occur in similar contexts have related meanings (Firth, 1957; Harris, 1954). Semantic relatedness allows us to measure the tendency for tokens to occur in coherent sets of words. Words that share strings of letters may be morphologically related (e.g., *teacher* and *teaching* share a morpheme) or etymologically related (e.g., *study* and *student* share a historical root), so we expect words that share tokens to be more closely related than randomly paired words, on average, which speaks to the semantic information that comes for free with word form (e.g., Dautriche et al., 2017). But previous work has emphasized the information in word form that tokens fail to capture, namely, morphology, so by comparing tokens to segments matched on length, we can investigate whether or in which languages byte-pair encoding captures more semantic information than random chunks of words do. In each of 41 languages, using all three tokenizers for each of the three conditions (tokens, segments matched on length, and the random baseline), we measured the semantic relatedness of 100,000 pairs of words. We then investigated how often those tokens correspond to morphemes and whether they impact LLM performance. All data and scripts are available at `https://osf.io/mzybx`.

2 Semantic Information in Tokens

2.1 Methods

2.1.1 *Materials*

We selected 41 languages with word-frequency ranks available from two databases, wordfreq (Speer, 2022) and FastText (Grave et al., 2018). We gathered words that are among the 50,000 most frequent words in both databases, which functions as a quality control and which ensures roughly equal frequency ranges for all languages ($M =$ 26,318.8 words per language; $SD =$ 5,623.7). This approach excludes very rare words, and subword tokens are especially important to representing rare words, but as we show below, most words in these samples comprise multiple tokens, so they allow us to investigate the tendency for tokens to occur in semantically related words without sacrificing quality. Along with the risk of junk among low-frequency words, distributional semantic models are less reliable for low-frequency words (e.g., Sahlgren and Lenci, 2016).

We tokenized the words in each language using OpenAI’s tiktoken Python package. For the token condition, we randomly paired each word with another word that shares a token, for each of its tokens, in each language using each tokenizer. For example, *baton* might be paired with *bathe* (which shares the GPT-4 token *bat*) and with *atonement* (which shares the GPT-4 token *on*). We removed any self-comparisons (e.g., *baton* paired with *baton*) and then randomly selected 1,000 pairs of words. We repeated this process 100 times, for 100,000 pairs in each LLM in each language. For the random baseline condition, we randomly paired words regardless of tokens, removed self-comparisons, randomly selected 1,000 pairs, and repeated this process 100 times. To compare the semantic information captured by tokens to the semantic information latent in word form, we randomly shuffled the lengths of the tokens in each word, randomly assigned shuffled lists of lengths to words of the same total length, and then segmented words into the assigned lengths. For example, *giraffes* has GPT-4 token lengths of three letters (*gir*), three letters (*aff*), and two letters (*es*); after shuffling, these lengths might be three letters, two letters, and three letters; and both *giraffes* and *cheetahs* are eight letters long, so if *cheetahs* were randomly assigned the shuffled list of lengths in *giraffes*, it would be split into segments of three letters (*che*), two letters (*et*), and three letters (*ahs*). As above, we randomly paired words that share length-matched segments, removed self-comparisons, selected 1,000 pairs, and repeated the process 100 times.

Byte-pair encoding complicates this procedure. First, GPT models incorporate whitespace into the first token of a word, and because the beginnings, middles, and ends of words convey different information (e.g., Cutler et al., 1985; St. Clair et al., 2009), this onset-marking is valuable. For example, the GPT-4 token *_bat* occurs in *_baton* and *_bathe* (the underscore represents whitespace), whereas *bat* occurs in *_verbatim* and *_subathing*. We therefore added whitespace before tokenizing words, to mimic how they would be encountered in sentences. In the length-matched condition, we coded whether each word incorporates whitespace into its first token, shuffled that coding across all words, and then did or did not mark word-initial segments accordingly, thereby matching the number of onset-marked words per condition. However, two of the languages do not include whitespace between words (Chinese and Japanese), so we did not add whitespace or onset markers in those cases. Four of the languages are written from right to left (Arabic, Hebrew, Persian, and Urdu), but as mentioned in Table 1, the GPT tokenizers incorporate leftward whitespace into the rightmost (word-initial) token, so we did not need to modify the procedure in these cases.

Second, GPT models operate on byte-level representations, and in many cases, especially languages that do not use the Latin alphabet, tokens are combinations of bytes that lack surface-level representations. They are instead represented as replacement characters, which look like question marks or empty squares. Despite having indistinguishable surface-level representations, these replacement characters correspond to different tokens with different embeddings, and they can be differentiated by using byte-level representations. Above, we described measuring and shuffling the lengths of tokens in letters, but in fact, we measured and shuffled lengths in bytes (hexadecimal digits).

2.1.2 *Measuring Semantic Relatedness*

We quantified semantic relatedness using FastText, with embeddings for 157 languages (Grave et al., 2018), and Polyglot, with word2vec embeddings for 137 languages (Al-Rfou et al., 2013). Embeddings are coordinates in semantic space, and both models situate words in 300-dimensional spaces determined by the language-specific contexts where words occur, such that words with related meanings are neighbours in semantic space. Using the Scikit-learn library (Pedregosa et al., 2011), we measured proximity in semantic space as cosine similarity. The cosine similarity of X and Y is the normalized dot product of X and Y:

$$K(X,Y) = \frac{\langle X, Y \rangle}{\|X\| \|Y\|}$$

FastText is known for having semantic representations of both words and subwords (e.g., *enc* in *encodings*), which seems ideal for measuring the information in tokens, but again, tokens lack surface-level representations in many cases, so they lack semantic representations in FastText.

FastText incorporates representations of subwords into word embeddings, which may inflate the cosine similarity of pairs that share tokens over the baseline condition, so after measuring the similarity of FastText embeddings, we repeated the process using the word2vec embeddings (with different pairs of words because Polyglot has a smaller vocabulary in some cases). Unlike FastText, word2vec has many negative cosine similarities in several languages. Negative cosine similarities are hard to interpret because they should indicate diametrically opposed meanings, but in practice, antonyms tend to have high positive cosine similarity (e.g., Ono et al., 2015), and negative cosine similarity instead seems to identify unrelated words that would be expected to have orthogonal embeddings, with cosines close to zero. So, following Günther et al. (2015) and Rotaru et al. (2018), we reset negative cosines to zero. As we show in the online supplement, resetting negative cosines substantially increases the correlation between FastText and word2vec when measuring of the amount of semantic information available in tokens (in one case, a correlation of .86 rather than .46), and it substantially increases the amount of variance explained in linear regression models (in one case, an adjusted R^2 of .55 rather than .29).

Input embeddings in LLMs (i.e., decontextualized representations of tokens) are comparable to embeddings from FastText and word2vec, but we did not use those representations. The reason for this is simple and essential: We are interested in words that comprise multiple tokens, such as *encodings* and *baton*, and LLMs do not have embeddings for any of those words. One way to represent the meanings of words that comprise multiple tokens would be, following Cassani et al. (2023) and others, to average the embeddings of their constituent tokens, but this would make the cosine similarity of words that share tokens trivial and circular. Furthermore, this approach would entrench the semantic impact of English. For example, combining the input embeddings of *lap+in* (the French word for ‘‘rabbit’’) would tell us little about the meaning ‘‘rabbit’’ and plenty about sitting in laps. The contextualized representations of those tokens, in higher layers, will indicate (in some sense) whether they are in a French context. Our question is whether those tokens tend to occur in other French words related to ‘‘rabbit’’, making it feasible that they play a role in representing that meaning rather than the meaning of that combination of tokens being learned through brute force.

One limitation of these embeddings is that words which occur in similar contexts might not be synonymous or substitutable (e.g., one of the nearest FastText neighbours of *encodings* is *multi-byte*). Human-annotated semantic similarity ratings have been used to address this shortcoming (Hill et al., 2015), and Multi-SimLex provides such ratings for 1,888 pairs of words from each of 13 languages (Vulić et al., 2020). We again investigated whether words that share tokens are more semantically similar than expected by chance or expected given comparable overlaps in word form. However, the pairs are not matched across our three conditions, and in some cases, very few pairs share tokens (e.g., English is mostly single-token words). We therefore focus on the similarity of FastText and word2vec embeddings and treat Multi-SimLex ratings as supplementary.

Finally, we measured how often pairs in each condition share a morpheme, as automatically parsed by Morfessor 2.0, also available from Polyglot (Creutz and Lagus, 2002; Kohonen et al., 2010; Virpioja et al., 2013). We compared the proportion of pairs that share morphemes in each condition, and we compared the semantic relatedness of pairs in each condition when they don't share any morphemes. Previous research has emphasized whether tokens align with morphemes, but there are many other regularities in form and meaning that can help LLMs represent rare words, such as the historical relationship between *purse* and *bursary*. Note, though, that Morfessor is a semi-supervised algorithm that only imperfectly approximates morphological segmentation, so any tendency for tokens to identify semantically related words that share no morphemes could, in theory, indicate cases where byte-pair encoding outperforms Morfessor at parsing words into morphemes, rather than cases where byte-pair encoding identifies extra-morphological regularities in form and meaning.

2.2 Results

2.2.1 *Multi-token Words per Language*

Do the representations of multi-token words matter? This depends, in part, on how common multi-token words are. We found that in every language other than English, most words in our sample comprise multiple tokens when represented by GPT-2, GPT-4, and GPT-4o, consistent with the fact that languages other than English use more GPT-4 tokens per word meaning (Petrov

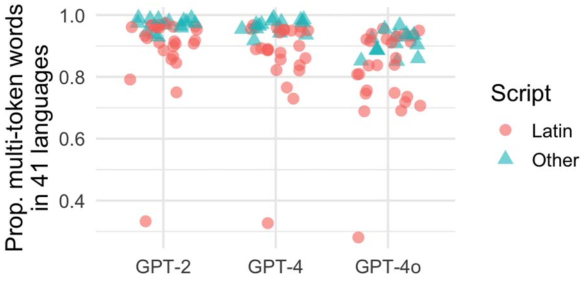

Figure 1: Proportion of words that comprise multiple tokens in each of 41 languages. The outlier for each tokenizer is English, where only about one third of words in our sample comprise multiple tokens.

et al., 2024). As we illustrate in Figure 1, the proportion of multi-token words is greatest in languages that use a script other than the Latin alphabet, and *t*-tests indicate that this difference is significant for all three tokenizers (all *p*-values < .001; see the online supplement).

2.2.2 *Semantic Information by Condition*

Do tokens imply word meanings? As illustrated in Figure 2, we found that pairs which share tokens (e.g., *baton* and *bathe*) are more semantically similar than randomly paired words (e.g., *baton* and *dog*) and are more semantically similar than pairs which share strings of bytes matched on length with tokens (e.g., *baton* and *glutton*), for all three tokenizers, using both FastText and word2vec embeddings. These differences are highly significant in paired *t*-tests (all *p*-values < .0001), with 41 observations per condition (i.e., the mean cosine similarity per language for each tokenizer in each condition).

Consistent with the FastText and word2vec findings, Multi-SimLex ratings are higher for pairs that share tokens than for pairs that do not, and this difference is highly significant for all three tokenizers in paired *t*-tests, with 13 observations per condition (because Multi-SimLex provides ratings for 13 languages; all *p*-values < .0001). In a follow-up analysis, we compared only pairs that are roughly matched on overlaps in word form, falling within the first and third quartiles of normalized edit distance in the token condition (i.e., excluding pairs that share many or few bytes; see the online supplement). Again, pairs that share tokens receive significantly higher semantic similarity ratings than pairs that do not share tokens. Overall, cosine similarity of FastText and word2vec embeddings and human-annotated semantic similarity ratings all suggest that byte-pair encoding segments words into strings which convey more information than would be expected in random chunks, so they may help LLMs represent word meanings.

2.2.3 *Semantic Information and Morphology*

How much of the semantic similarity of pairs that share tokens should be attributed to morphology? As illustrated in Figure 3, we found that pairs which share tokens more often share morphemes, as parsed by Morfessor, than pairs which share length-matched segments do and more often than randomly paired words do. Paired *t*-tests of the proportion of shared morphemes by condition, for each tokenizer in each language, indicate that the differences between the token condition and the control conditions are highly significant (all *p*-values < .0001; see the online supplement). However, among pairs that share both tokens and morphemes, tokens often differ from morphemes: They have the same surface forms only 42% of the time for GPT-4o, 32% for GPT-4, and 21% for GPT-2. This disparity between the proportion of pairs that share morphemes and the occasions in which those tokens look like morphemes suggests that past studies' emphasis on morphology might gloss over the success of seemingly malformed tokens in identifying morphemes (e.g., chunks such as *enc* in *encodings* and *encodified*).

Furthermore, when analyzing pairs that do not share any morphemes, pairs that share tokens are still more semantically similar than control pairs. Paired *t*-tests of mean semantic similarity by condition, for each tokenizer in each language, are significant for the FastText and word2vec embeddings for all three tokenizers (all *p*-values < .0001). The differences are not significant for the Multi-SimLex human ratings, but those ratings trend in the expected direction and likely fail to reach significance because only 11 languages have ratings in all conditions. As we show in the online supplement, unpaired *t*-tests on trial-level data (i.e., a separate data point for each pair of words in each language) find that pairs which share tokens but not morphemes receive significantly higher ratings than pairs which share neither tokens nor morphemes, for all three tokenizers (all *p*-values < .0001). This suggests that not only do tokens successfully identify morphemes, even when they don't neatly map onto

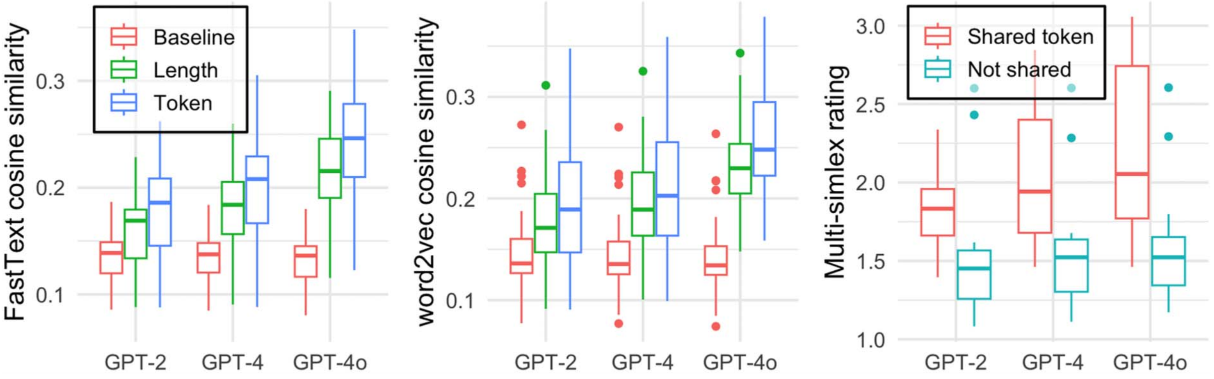

Figure 2: The mean semantic similarity of 100,000 pairs of words in each of 41 languages (or 1,888 pairs in 13 languages for Multi-SimLex) which share tokens, share length-matched segments, or are randomly paired.

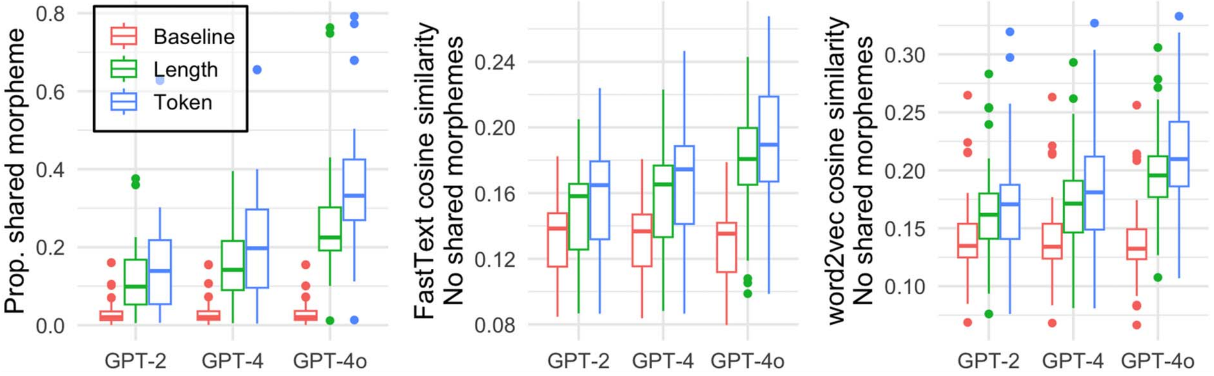

Figure 3: Proportion of shared morphemes, and semantic similarity among pairs that share no morphemes.

those morphemes, but that tokens also identify other patterns in form and meaning, such as the etymological relationship between *purse* and *bursary*. Tokenization methods that slavishly conform to morphology would obscure many such patterns. However, as mentioned above, these morphemes are identified by a semi-supervised algorithm, not by painstaking human annotation, which limits claims about morphology. An alternative explanation is that byte-pair encoding outperforms Morfessor at identifying morphemes, consistent with the broader conclusion that tokens capture valuable semantic information.

2.2.4 *Semantic Information by Language*

When averaging across all 41 languages, words that share tokens are more semantically similar than words that share length-matched segments, but the amount of semantic information available in tokens varies by language. We therefore conducted paired *t*-tests of semantic similarity in the token condition versus the control conditions, separately for each language, with a separate data point for each pair of words. For the GPT-2 tokenizer, only the 26 languages which use the Latin alphabet have significantly more similar FastText or word2vec embeddings in the token condition than in the length-matched condition. For GPT-4, only the 30 languages that use either the Latin or Cyrillic script have significantly more similar FastText or word2vec embeddings in the token condition than in the length-matched condition—though Macedonian, which uses Cyrillic, does not reach significance for FastText embeddings. For GPT-4o, only Greek and Japanese do not have significantly more similar FastText embeddings in the token condition than in the length-matched condition, and only Greek, Japanese, Chinese, and Korean do not have significantly more similar word2vec embeddings. The difference between the token condition and the random baseline fails to reach significance only when measuring word2vec embeddings of

Tamil words that share GPT-2 tokens. For most languages, pairs do not receive significantly higher Multi-SimLex ratings when they share tokens, but with few observations in unpaired *t*-tests, there is little to be inferred from these null effects. For instance, English fails to reach significance because it has only five or six Multi-SimLex pairs that share tokens for each tokenizer, out of 1,888.

What language-level variables might explain the amount of semantic information conveyed by tokens? Different languages have different FastText and word2vec semantic spaces, with uninterpretable dimensions, which precludes direct comparisons of cosine similarity across languages. Instead, to quantify how reliably tokens imply word meanings, we divided the mean cosine similarity of word embeddings in the token condition by the mean in the random baseline and by the mean in the length-matched condition, for each language using each tokenizer. This indicates how much more (or less) information tokens capture, on the scale of cosines in those other conditions. For example, FastText embeddings for pairs of Tagalog words have a mean cosine similarity of .201 when they share GPT-4 tokens, .178 when they share segments matched on length with GPT-4 tokens, and .135 in the baseline condition, for metrics of 1.13 over length-matched segments and 1.49 over the random baseline. (A value of 1.0 would indicate no value added over that control condition.)

We considered two language-level predictors of the semantic information available in tokens. First, we quantified how high resource a language is as its percentage of the Common Crawl (i.e., how much of the internet it makes up). English makes up about 46% of the Common Crawl; the second-highest resource language, German, makes up less than 6%; and the lowest resource language in our sample, Tagalog, makes up less than 0.01% (per CC-MAIN-2023-23). To capture linear relationships, we log-transformed these percentages.

Second, we quantified a language's orthographic similarity to English as the normalized edit distance between words in Swadesh lists, available from the NLTK library (Bird et al., 2009). The Swadesh lists comprise 207 core vocabulary items in hundreds of languages, and when languages are more closely related, words with the same meaning are more likely to have similar spelling (which is more relevant to LLMs than phonological similarity). Following Kumar et al. (2022), we divided the absolute edit distance between A and B by the length of the longer of those two words:

$$D(A, B) = \frac{E(A, B)}{max(Len_A, Len_B)}$$

For example, the German word *und* has an edit distance of 1 from its English translation, *and*, because you need to add, remove, or swap one letter to make them identical. Normalized for length (i.e., divided by the maximum number of edits possible), *und* and *and* have an edit distance of .33, differing in one out of three letters. Across the 207 words, German has a mean distance of .69 from English, French has a distance of .75, Turkish has a distance of .89, and the 15 languages which use a script other than the Latin alphab et all have the maximum distance of 1. We considered also grouping languages by family (e.g., Hebrew and Arabic are Semitic) or by script family (e.g., Russian, Ukrainian, Macedonian, and Bulgarian use Cyrillic), but including factors with many unordered levels, some of which correspond to only one observation (e.g., Tamil and Chinese), led to overfitting and was uninformative.

In four linear regression models (two for FastText embeddings and two for word2vec), we regressed the semantic information available in tokens over the control conditions (two models for the random baseline and two for length-matched segments) on tokenizer (sum-coded, to reflect the increasing vocabulary size and increasing multilingualism of GPT-4 over GPT-2 and GPT-4o over GPT-4) and its interaction with Common Crawl percentage (log-transformed) plus its interaction with orthographic distance from English. Because English is an outlier in terms of Common Crawl percentage and distance from itself, we excluded it from these analyses, giving us 120 observations per condition (40 languages segmented by each of three tokenizers). In all four models, we found a significant main effect of tokenizer, such that tokens convey more information in LLMs with larger, more multilingual vocabularies, relative to length-matched segments and the random baseline. See Figure 4 and Table 2.

Common Crawl percentage is a significant positive predictor of the semantic information conveyed by tokens over the random baseline, for both FastText and word2vec, indicating that

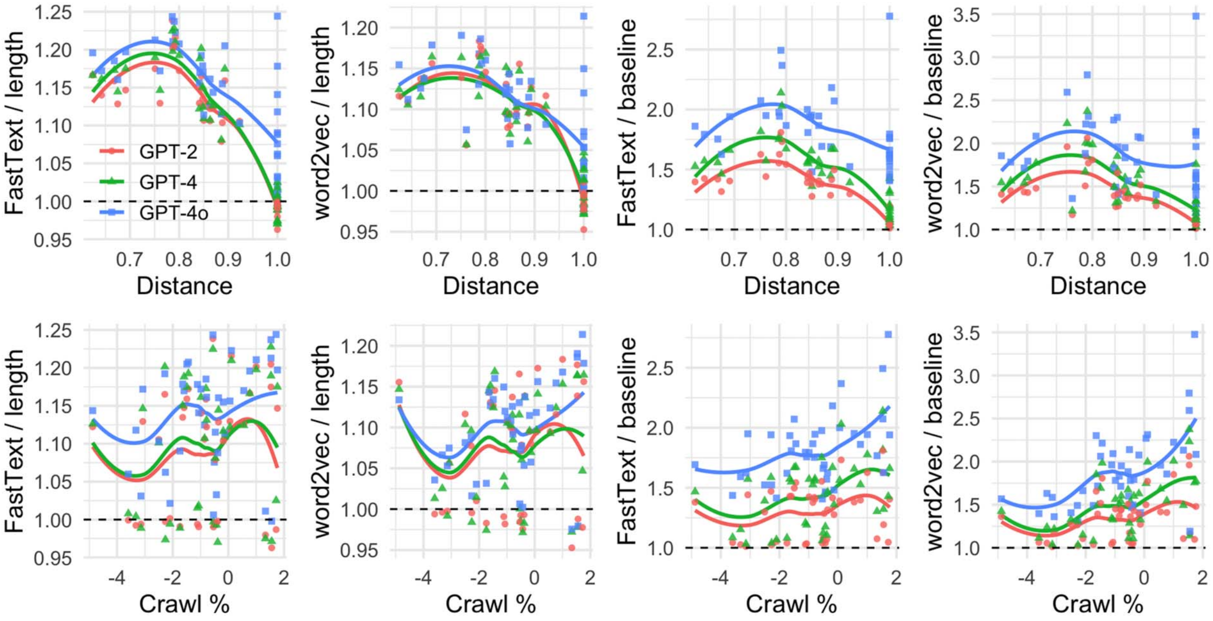

Figure 4: The mean cosine similarity of words that share tokens divided by the mean similarity of words that share length-matched segments or by the mean similarity of randomly paired words, in 40 languages, as a function of a language's orthographic distance from English (top row) and the percentage of the Common Crawl it makes up (log-transformed, bottom row). The dashed black line indicates the at-chance level, when the mean similarity in the token condition is the same as in the length-matched condition or the random baseline. The smoothed lines (using GAM) are for illustration only.

	FastText token / length			word2vec token / length			FastText token / baseline			word2vec token / baseline		
	β	*SE*	*p*	β	*SE*	*p*	β	*SE*	*p*	β	*SE*	*p*
Intercept	1.12	.00	<.001	1.08	.00	<.001	1.52	.01	<.001	1.57	.01	<.001
Tokenizer	0.05	.01	<.001	0.03	.01	<.001	0.52	.03	<.001	0.53	.04	<.001
Distance	−0.06	.00	<.001	−0.05	.00	<.001	−0.14	.01	<.001	−0.12	.02	<.001
Crawl %	−0.00	.00	.693	−0.00	.00	.975	0.07	.01	<.001	0.15	.02	<.001
Tok.*Dist.	0.02	.01	<.001	0.03	.01	<.001	0.10	.03	<.001	0.15	.04	<.001
Tok.*CC%	0.01	.01	.166	0.01	.01	.022	0.10	.03	<.001	0.18	.04	<.001
Adj. R^2			.651			.581			.640			.554

Table 2: Cosine similarity of embeddings for words that share tokens divided by the similarity of words that share length-matched segments or randomly paired words (the baseline), regressed on tokenizer (sum-coded: GPT-2 = −0.5, GPT-4 = 0, GPT-4o = 0.5), Common Crawl percentage (log-transformed and z-scored), and orthographic distance from English (z-scored).

higher resource languages have more informative tokens. The interaction of Common Crawl percentage with tokenizer is a significant positive predictor of FastText and word2vec similarity over the random baseline, suggesting that the advantage for higher resource languages *increases* in GPT-4o over GPT-4 and GPT-2, rather than GPT-4o mitigating disadvantages for lower resource languages. This appears to be due to higher resource languages having longer tokens, as Common Crawl percentage is not a significant predictor of FastText or word2vec similarity in the token condition over the length-matched condition.

Orthographic distance from English is a significant negative predictor in all four models, indicating that tokens convey less information in languages that differ more from English, and the interaction with tokenizer is significantly positive in all cases, indicating that GPT-4o tokens mitigate the disadvantage of dissimilarity to English.

Much of these effects might stem from tokens conveying less information in languages which don't use the Latin alphabet (i.e., which differ from English in all characters). When analyzing only the 25 languages other than English that use the Latin alphabet, orthographic distance is a significant negative predictor of token information over the length-matched condition but not over the random baseline. In fact, when predicting information over the random baseline, distance from English trends in the opposite direction, and it is a significant *positive* predictor when measuring the similarity of word2vec embeddings. The effect of orthographic distance from English is mostly a matter of whether a language uses the Latin alphabet.

3 GPT-4 Benchmark Performance

Do LLMs use the semantic information available in tokens to represent word meanings? OpenAI reports GPT-4's performance on the massive multi-task language understanding benchmark for 27 languages (MMLU; Hendrycks et al., 2020), so we investigated whether the semantic information available in those languages' GPT-4 tokens predicts MMLU scores. Nine of the 27 languages have word-frequency ranks available from FastText but not wordfreq (i.e., they're not among our original 41 languages), so we gathered the 26,201 most-frequent words in each of those nine languages (according to FastText word frequency ranks), which is the median number of words among the 50,000 most-frequent words in both databases for the other 41 languages. We followed the same procedure of measuring the cosine similarity of FastText and word2vec embeddings for 100,000 pairs of words in each of three conditions for each language. To quantify the semantic information available in tokens for each language, we divided the mean cosine similarity of word embeddings for pairs that share tokens by the mean similarity of randomly paired words, according to FastText and word2vec, as above. We ignored the length-matched condition because when it comes to LLM performance, we care only about the amount of information available in tokens, not whether that information is stumbled across by virtue of length. That is, penalizing languages for having long tokens will obscure any impact of token informativity. Multi-SimLex provides ratings for only eight of the 27 languages with MMLU scores. As we report in the online supplement, the effects are consistent with FastText and word2vec, but the adjusted R^2 is over .99 because the number of parameters in the models described below is almost equal to the number of Multi-SimLex observations, so the model is clearly overfit.

	FastText		word2vec	
	β	p	β	p
Intercept	77.7	<.0001	79.5	<.0001
Crawl %	3.7	<.0001	2.8	.0020
Distance	−1.8	.1423	−2.0	.0390
Info.	1.0	.4110	2.3	.0530
Info.*CC%	−2.5	.0070	−2.2	.0227
Info.*Dist.	−1.5	.3610	0.5	.6328
Adj. R^2		.699		.700

Table 3: GPT-4 MMLU score regressed on Common Crawl percentage (log-transformed and z-scored), orthographic distance from English (z-scored), and semantic information available in GPT-4 tokens (i.e., mean cosine similarity in the token condition divided by mean similarity in the baseline condition, z-scored).

In two linear regression models, for FastText and word2vec embeddings, we regressed MMLU score on the interaction of semantic information available in GPT-4 tokens with the percentage of the Common Crawl that a language makes up plus its interaction with orthographic distance from English. We again excluded English because it is an outlier. In both cases, as reported in Table 3, Common Crawl percentage is a significant positive predictor, such that GPT-4 performs better on higher resource languages. The positive main effect of semantic information conveyed by tokens is not significant in either model, but as illustrated in Figure 5, the interaction with Common Crawl percentage is significantly negative, which suggests that GPT-4 relies less on meaningful constituents when it has sufficient training data (i.e., it engages in inference more when memorization isn't feasible). Most notably, Afrikaans makes up less than 0.01% of the Common Crawl, but it has more semantic information available in its tokens than Chinese does and almost as much as Russian does, and GPT-4's MMLU score is higher for Afrikaans than for Chinese or Russian.

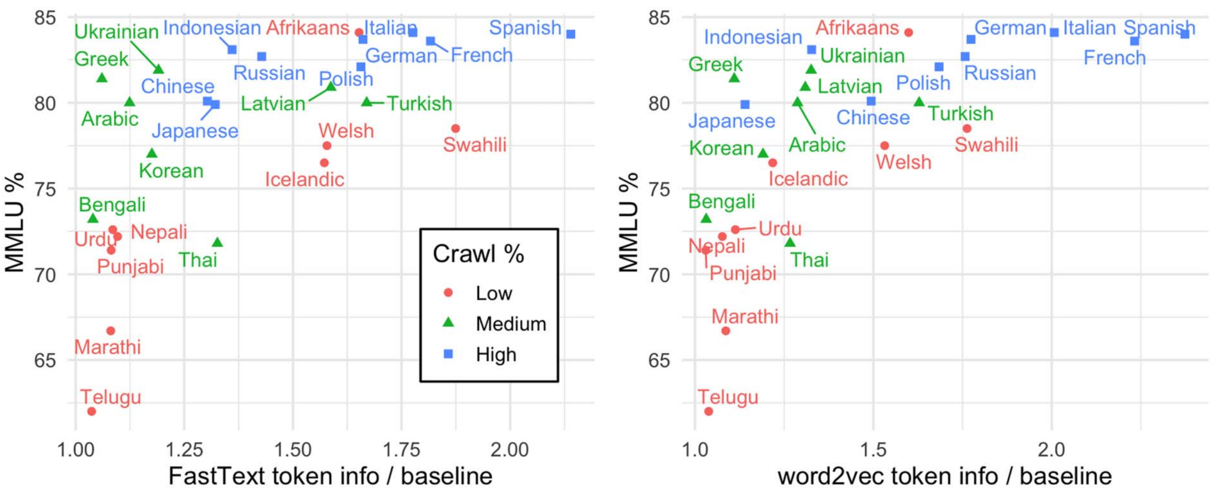

Figure 5: GPT-4 benchmark performance as a function of the information available in GPT-4 tokens and Common Crawl % (split into three bins for the sake of illustration). GPT-4 performs well on high-resource languages regardless of token informativity, but performance plummets on low-resource languages with uninformative tokens.

When fitting the models without orthographic distance from English, the main effect of information in tokens is significantly positive, as we show in the online supplement. However, vice versa is also true: When removing token informativity from the models, distance from English is significantly negative. Lower resource languages seem to benefit from informative tokens, but because the information available in tokens correlates with orthographic distance from English, this could be a spurious effect better attributed to the transfer of information between etymologically related languages (cf. Pires, 2019) or to an advantage for languages that use the Latin alphabet (cf. Ahuja et al., 2023). For example, Afrikaans has informative tokens, but as a Germanic language, it is also closely related to English. Further confounding these effects, languages that make up more of the training data will tend to have longer tokens (because those sequences of bytes will be more common), so, assuming that longer tokens are more informative, the information available in tokens will correlate with languages having more opportunities for memorization (i.e., not needing to infer meaning from constituents). This is only partially controlled for by including Common Crawl percentage as a covariate, since the Common Crawl only roughly approximates how we expect language proportions to break down in GPT-4's proprietary training data. We therefore provide only suggestive, correlational evidence that informative tokens improve performance on lower resource languages.

4 General Discussion and Conclusion

We compared the semantic relatedness of words that share tokens, such as *bat+on* and *bat+he*, to words that share length-matched segments, such as *bat+on* and *gl+utton*, and to randomly paired words. Words that share tokens are more closely related than words in the control conditions; words that share tokens more often share morphemes than words in the control conditions do, even when the tokens are not themselves morphemes; and when words that share tokens don't share any morphemes, they are still more closely related than words in the control conditions. These findings suggest that standard frequency-based tokenization (i.e., byte-pair encoding) successfully picks out meaningful constituents, that those tokens don't need to look like morphemes in order to identify morphological patterns, and that frequency-based tokenization recovers other sorts of patterns, such as etymological relationships (e.g., *purse* and *bursary*).

Of course, doing better than chance in no way entails optimal segmentation, and the shortcomings of GPT tokens are most obvious in languages that are orthographically dissimilar to English. GPT-4o mitigates this problem, segmenting words that use non-Latin scripts into more meaningful tokens than GPT-4 does, yet GPT-4o also amplifies the disadvantage for lower resource languages (i.e., those which make up less of the Common Crawl). We've provided some evidence that LLMs use the information available in tokens to compensate for the challenge of processing low-resource languages: GPT-4 performs better on lower resource languages that have more meaningful tokens, but the corollary is that performance drops on lower resource languages with uninformative tokens. It remains to be seen whether this pattern of effects holds in GPT-4o and other LLMs and when controlling for potential confounds.

Reliable representations of word meanings matter because LLMs are black boxes, so content moderation depends on natural language. For example, when the text-to-image model DALL-E generates images, OpenAI appends user-generated prompts with system prompts such as ‘‘Do not create any imagery that would be offensive’’ and ‘‘Do not discuss copyright policies’’ (github.com/spdustin). Engineers exert some control over LLM output through the curation of training data and through reinforcement learning from human feedback (e.g., Ouyang et al., 2022), but the most direct method of content moderation involves talking to LLMs and assuming they interpret words like people do. Piantadosi and Hill (2022) argue that LLMs can learn human-like meanings because ‘‘words and concepts must be used jointly together in a coherent way that mimics how humans would.’’ Patterns in words do convey valuable semantic information, but recall our first, simplest finding: that most words in every language other than English comprise multiple tokens. Even English, in a larger sample of the 57,531 words which occur in multiple films and TV shows in the SUBTLEX-US database (Brysbaert and New, 2009), follows this trend, with two thirds of those words being segmented into multiple tokens by GPT-2, GPT-4, and GPT-4o. The outsize proportion of multi-token words belies common descriptions of LLMs as learning patterns in words by predicting the next word in a sequence (e.g., Bubeck et al., 2023). LLMs clearly do represent the meanings of many multi-token words (e.g., Kaplan et al., 2024), but again, most words are rare, and interpreting rare words reliably and in a way that aligns with human interpretations requires inference based on word form. We’ve found that in many languages, subword tokens occur in semantically coherent sets of words, creating opportunities for human-like inference, yet token utility decreases when it’s needed most, in lower resource languages. If LLMs continue to carve those words far from their joints, disparities in performance will persist.

5 Limitations

This study relies heavily on distributional semantic models, such as word2vec, which measure the tendency for words to occur in similar contexts and so provide a narrow window onto word meaning. Words with similar distributions are often thematically related without sharing semantic features (e.g., Erk, 2016). For example, *dog* and *leash* co-occur often, but unlike *dog* and *cat*, they are taxonomically distant. Even words with opposite meanings tend to have similar distributions (e.g., Ono et al., 2015). So, a token which occurs in a coherent set of contexts won’t always reliably imply word meaning. We corroborated distributional similarity with ratings from humans who were instructed to focus on semantic features rather than co-occurrence, and we further found that words which share tokens often share morphemes, which tend to convey semantic features (though we used Morfessor 2.0 to imperfectly identify morphemes, which is another limitation). These supplementary analyses support our conclusion that GPT tokens imply word meanings better than random chunks of words do, but this study nevertheless exemplifies how the availability of distributional semantic models biases research. Our conclusions would be strengthened by convergent evidence using alternative measures of word meaning.

Acknowledgments

We owe many thanks to three anonymous reviewers and to the TACL action editor, Nathan Schneider. DAH was supported by a PhD Fellowship from the Hong Kong Research Grants Council.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. Mega: Multilingual evaluation of generative ai. *arXiv preprint arXiv:2303.12528*. https://doi.org/10.18653/v1/2023.emnlp-main.258

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*.

Maria Alegre and Peter Gordon. 1999. Frequency effects and the representational status of regular inflections. *Journal of Memory and Language*, 40(1):41–61. https://doi.org/10.1006/jmla.1998.2607

Catherine Arnett, Pamela D. Rivière, Tyler A. Chang, and Sean Trott. 2024. Different tokenization schemes lead to comparable performance in spanish number agreement. *arXiv preprint arXiv:2403.13754*. https://doi.org/10.18653/v1/2024.sigmorphon-1.4

Rolf Harald Baayen. 2012. *Word Frequency Distributions*, volume 18. Springer Science & Business Media.

Rolf Harald Baayen, Yu-Ying Chuang, Elnaz Shafaei-Bajestan, and James P. Blevins. 2019. The discriminative lexicon: A unified computational model for the lexicon and lexical processing in comprehension and production grounded not in (de) composition but in linear discriminative learning. *Complexity*, 2019(1):4895891. https://doi.org/10.1155/2019/4895891

Khuyagbaatar Batsuren, Ekaterina Vylomova, Verna Dankers, Tsetsuukhei Delgerbaatar, Omri Uzan, Yuval Pinter, and Gábor Bella. 2024. Evaluating subword tokenization: Alien subword composition and oov generalization challenge. *arXiv preprint arXiv:2404.13292*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146. https://doi.org/10.1162/tacl_a_00051

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. *arXiv preprint arXiv:2004.03720*. https://doi.org/10.18653/v1/2020.findings-emnlp.414

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior Research Methods*, 41(4):977–990. https://doi.org/10.3758/BRM.41.4.977, PubMed: 19897807

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv: 2303.12712*.

Giovanni Cassani, Fritz Günther, Giuseppe Attanasio, Federico Bianchi, and Marco Marelli. 2023. Meaning modulations and stability in large language models: An analysis of bert embeddings for psycholinguistic research. *osf.io/preprints/psyarxiv/b45ys*. https://doi.org/10.31234/osf.io/b45ys

Kenneth Ward Church. 2020. Emerging trends: Subwords, seriously? *Natural Language Engineering*, 26(3):375–382. https://doi.org/10.1017/S1351324920000145

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. *arXiv preprint cs/0205057*. https://doi.org/10.3115/1118647.1118650

Anne Cutler, John A. Hawkins, and Gary Gilligan. 1985. The suffixing preference: A processing explanation. *Linguistics*. https://doi.org/10.1515/ling.1985.23.5.723

Isabelle Dautriche, Kyle Mahowald, Edward Gibson, and Steven T. Piantadosi. 2017. Wordform similarity increases with semantic similarity: An analysis of 100 languages. *Cognitive Science*, 41(8):2149–2169. https://doi.org/10.1111/cogs.12453, PubMed: 27862241

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mark Dingemanse, Damián E. Blasi, Gary Lupyan, Morten H. Christiansen, and Padraic Monaghan. 2015. Arbitrariness, iconicity, and systematicity in language. *Trends in Cognitive*

Sciences, 19(10):603–615. https://doi.org/10.1016/j.tics.2015.07.013, PubMed: 26412098

Katrin Erk. 2016. What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, 9:17–1. https://doi.org/10.3765/sp.9.17

Fengxiang Fan. 2010. An asymptotic model for the english hapax/vocabulary ratio. *Computational Linguistics*, 36(4):631–637. https://doi.org/10.1162/coli_a_00013

John Firth. 1957. A synopsis of linguistic theory 1930–1955. *Studies in Linguistic Analysis, Special Volume/Blackwell*.

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

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Fritz Günther, Carolin Dudschig, and Barbara Kaup. 2015. Lsafun-An R package for computations based on latent semantic analysis. *Behavior Research Methods*, 47(4):930–944. https://doi.org/10.3758/s13428-014-0529-0, PubMed: 25425391

Bernal Jimenez Gutierrez, Huan Sun, and Yu Su. 2023. Biomedical language models are robust to sub-optimal tokenization. *arXiv preprint arXiv:2306.17649*. https://doi.org/10.18653/v1/2023.bionlp-1.32

Zellig Harris. 1954. Distributional structure. *WORD*, 10(2–3):146–162. https://doi.org/10.1080/00437956.1954.11659520

David A. Haslett and Zhenguang G. Cai. 2023. Similar-sounding words flesh out fuzzy meanings. *Journal of Experimental Psychology: General*, 152(8):2359. https://doi.org/10.1037/xge0001409, PubMed: 37307335

David A. Haslett and Zhenguang G. Cai. 2024. Systematic mappings of sound to meaning: A theoretical review. *Psychonomic Bulletin & Review*, 31(2):627–648. https://doi.org/10.3758/s13423-023-02395-y, PubMed: 37803232

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695. https://doi.org/10.1162/COLI_a_00237

Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves bert's interpretation of complex words. *arXiv preprint arXiv:2101.00403*. https://doi.org/10.18653/v1/2021.acl-long.279

Valentin Hofmann, Hinrich Schuetze, and Janet B. Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Association for Computational Linguistics*. https://doi.org/10.18653/v1/2022.acl-short.43

Haris Jabbar. 2023. Morphpiece: Moving away from statistical language representation. *arXiv preprint arXiv:2307.07262*.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. 2024. From tokens to words: On the inner lexicon of llms. *arXiv preprint arXiv:2410.05864*.

Ayush Kaushal and Kyle Mahowald. 2022. What do tokens know about their characters and how do they know it? *arXiv preprint arXiv:2206.02608*. https://doi.org/10.18653/v1/2022.naacl-main.179

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86.

Abhilasha A. Kumar, Nancy B. Lundin, and Michael N. Jones. 2022. Mouse-mole-vole: The inconspicuous benefit of phonology during retrieval from semantic memory. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. https://doi.org/10.31234/osf.io/2bazx

Manuel Mager, Arturo Oncevay, Elisabeth Mager, Katharina Kann, and Ngoc Thang

Vu. 2022. Bpe vs. morphological segmentation: A case study on machine translation of four polysynthetic languages. *arXiv preprint arXiv:2203.08954*. https://doi.org/10.18653/v1/2022.findings-acl.78

Marco Marelli, Simona Amenta, and Davide Crepaldi. 2015. Semantic transparency in free stems: The effect of orthography-semantics consistency on word recognition. *Quarterly Journal of Experimental Psychology*, 68(8):1571–1583. https://doi.org/10.1080/17470218.2014.959709, PubMed: 25269473

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *arXiv preprint arXiv:2112.10508*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 984–989.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2024. Hello gpt-4o. Openai.com/index/hello-gpt-4o.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kyubyong Park. 2020. An empirical study of tokenization strategies for various korean nlp tasks. *arXiv preprint arXiv:2010.02534*. https://doi.org/10.18653/v1/2020.aacl-main.17

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2024. Language model tokenizers introduce unfairness between languages. *Advances in Neural Information Processing Systems*, 36.

Steven T. Piantadosi and Felix Hill. 2022. Meaning without reference in large language models. *arXiv preprint arXiv:2208.02957*.

Telmo Pires. 2019. How multilingual is multilingual bert. *arXiv preprint arXiv:1906.01502*. https://doi.org/10.18653/v1/P19-1493

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Armand S. Rotaru, Gabriella Vigliocco, and Stefan L. Frank. 2018. Modeling the structure and dynamics of semantic processing. *Cognitive Science*, 42(8):2890–2917. https://doi.org/10.1111/cogs.12690, PubMed: 30294932

Magnus Sahlgren and Alessandro Lenci. 2016. The effects of data size and frequency range on distributional semantic models. *arXiv preprint arXiv:1609.08293*. https://doi.org/10.18653/v1/D16-1099

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774. https://doi.org/10.1609/aaai.v34i05.6403

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint* *arXiv:1508.07909*. https://doi.org/10.18653/v1/P16-1162

Robyn Speer. 2022. rspeer/wordfreq: v3. 0. *Version v3. 0.2. Sept*.

Michelle C. St. Clair, Padraic Monaghan, and Michael Ramscar. 2009. Relationships between language structure and language learning: The suffixing preference and grammatical categorization. *Cognitive Science*, 33(7):1317–1329. https://doi.org/10.1111/j.1551-6709.2009.01065.x, PubMed: 21585507

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–21. https://doi.org/10.1145/3578707

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Ivan Vulić, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, et al. 2020. Multi-SimLex: A large-scale evaluation of multilingual and cross-lingual lexical semantic similarity. *Computational Linguistics*, 46(4):847–897. https://doi.org/10.1162/coli_a_00391

Yonghui Wu, Jiang Xu, Diogo Almeida, Carroll Wainwright, Pamula Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.