Abstract

Natural language processing systems often struggle with out-of-vocabulary (OOV) terms, which do not appear in training data. Blends, such as *innoventor*, are one particularly challenging class of OOV, as they are formed by fusing together two or more bases that relate to the intended meaning in unpredictable manners and degrees. In this work, we run experiments on a novel dataset of English OOV blends to quantify the difficulty of interpreting the meanings of blends by large-scale contextual language models such as BERT. We first show that BERT’s processing of these blends does not fully access the component meanings, leaving their contextual representations semantically impoverished. We find this is mostly due to the loss of characters resulting from blend formation. Then, we assess how easily different models can recognize the structure and recover the origin of blends, and find that context-aware embedding systems outperform character-level and context-free embeddings, although their results are still far from satisfactory.

1 Introduction

For the token-based architectures that dominate contemporary natural language processing, a particularly difficult form of linguistic generalization arises from unseen phenomena at the word level, where novel sequences of characters, morphemes, or phonemes are known as out-of-vocabulary (OOV) terms (Brants, 2000; Plank, 2016; Heigold et al., 2017). Pretrained transformers like BERT (Devlin et al., 2019) handle OOV terms by subtokenization: segmenting all whitespace-delimited tokens into smaller units, from which any OOV term can be constructed (Sennrich et al., 2016). But while this approach is well suited for phenomena like concatenative English morphology, many linguistic processes generate OOV terms that cannot be cleanly decomposed into meaningful subtoken segments.

In this paper we address a particularly interesting and challenging source of OOV terms: novel blends (Algeo, 1977), also known as portmanteaux (Deri and Knight, 2015a). Blends are constructed from the combination of multiple bases into a new form, in which some characters is shared across both bases: for example, *shop + optics* = *shoptics*. In this way, blends differ from other lexical compounds (e.g., *watermelon* = *wa + ter + melon*), which are formed by simple concatenation. Examples of OOV blends and their bases from our novel English blends dataset, collected from a natural source linked to the blends’ originating contexts (§2), are presented in Table 1. OOV blends are especially challenging to process, due to their combination of function-level semantic novelty with the form-level pathology of an unexpected character sequence.

Our main contribution is to offer what is to our knowledge the first analysis of how transformer-based contextual embedding models process novel blends and the representations they are able to produce for these challenging forms. First, we examine the impact of blends’ wordforms by comparing the ability of contextualized models to represent novel blends against the minimally-different case of novel lexical compounds. In §3, we show that the limited ability of contextual language models to represent blends against the minimally-different case of novel lexical compounds. In §3, we show that the limited ability of contextual language models to represent novel blends’ components faithfully is primarily attributable to their form properties, whereas semantic differences between compounds and blends play a much smaller role. We

---

1 Another approach is to operate directly at the character level (e.g., Ling et al., 2015), but this has not been widely adopted within the mainstream framework of transformer-based models (Vaswani et al., 2017), due in part to the difficulty of scaling transformer-based models to sufficiently large contexts when operating at the character level.
then investigate how well several methods are able to recover the morphological boundaries within blends, which could mitigate the impact by splitting blends into segments contributed by each constituent base (§4.1). Finally, we attempt to recover the constituent bases given a segmentation (§4.2). Even under favorable conditions, we find that systems proposed previously for similar tasks struggle on blends, showing limitations of form-based and distributional similarity approaches. We propose a novel unsupervised base recovery method using contextualized masked language models, BERT RANKER. While this system performs well relative to others, we find substantial room for improvement. In our view, these results demonstrate the need for future work on our novel dataset and associated tasks.2

2 Complex Words Dataset

Our proposed investigation of the behavior of NLP systems on novel complex words requires a high-quality, reliable resource of truly novel blends and compounds in their original contexts, annotated for character sequence composition and semantic properties. The NYTWIT dataset (Pinter et al., 2020) contains English words new to the New York Times extracted by a bot between the dates of November 2017 and March 2019 with associated news article contexts. Words were annotated for their type of novelty.

We extract and further annotate three types from this dataset (version 1.1): blends (142 items), transparent compounds (121), and opaque compounds (49).4 The difference between the compound classes is semantic and somewhat subjective: transparent compounds have meanings which are comprehensible with little context (e.g. quiz-maker, a person who makes quizzes), while opaque compounds exhibit metaphoric or allusive semantics (e.g. deathbox, a dangerous car).

The first two authors annotated each word for its constituent bases, the character locations in which each base is represented, and the semantic relation between the bases. A sample of annotated blends is presented in Table 1. All disagreements resulting from the first round of blend base annotation (7%) were resolved by discussion, with the help of the words’ originating context. These contexts vary considerably in their length and informativity,5 but typically contain direct or indirect disambiguating information, and sometimes the component bases themselves: for blends, 40.3% of documents contained at least one of the bases within sentences where the blends appear (e.g. only shop or optics, for shoptics), while 10.2% contained both.

Semantic relations. An author annotated all blends and compounds in the dataset according to the well-studied semantic taxonomy of Tratz and Hovy (2010), which was designed for multiword nominal compounds (e.g. cooking pot). These relations were not intended to be applied to other types of phrases, blends, or lexical compounds. However, by referring to the official taxonomy, the expanded definitions in Dima and Hinrichs (2015), and the coarse- and fine-grained relation training data, the annotator was able to assign one of twelve coarse-grained relation classes to each word.6

As a preliminary check, we trained a relation classifier following the approach of Dima and Hinrichs (2015), a single-hidden-layer classifier over GloVe embeddings (Pennington et al., 2014), on the RANDOM partition of the Tratz and Hovy

2We release our code and data at http://github.com/yuvalpinter/unblend.
3www.twitter.com/NYT_First_Said
4Originally annotated as “compositional” and “new” compounds, respectively.
5Compare, for example, the following context sentences: “Blaspy?”; “The procrastibaker must believe that it is possible to be simultaneously working on a document, buttering pans and separating eggs.”
6For example, ATTRIBUTE was applied to the adjective-noun blend fitfluencers. 69% of the words contain a non-noun base.
Novel blends are a unique linguistic phenomenon, posing challenges for automated systems on many different levels. However, the sparsity of their appearances in real-world text, as well as the expertise required for creating a natural language understanding task which uses specific documents from a large variety of domains as supporting information, make the evaluation of the effect of novel blends on this type of downstream task an impractical goal for the scope of this work. Instead, we assess the treatment of novel blends at the representational step of contemporary contextualized language models, by performing an analysis of their processing by BERT (Devlin et al., 2019). To gauge how well BERT represents blends, we conduct a comparison with its treatment of a minimally-different control class of novel words, namely lexical compounds. These are forms where at least two bases are concatenated in full (e.g. quizmaker), without the character loss incurred in blends.

Our analysis begins with the assumption that in any given context, the meaning representation of a complex word (blend or lexical compound) must be composed from its bases, which we can estimate using the representational similarity between a complex word and its bases in the same context. This criterion can be viewed as a form of linguistic generalization, and if satisfied, enables downstream models to produce consistent results across related words. To test this criterion, we compute the vector similarities between the contextualized representations of complex words and their components, a method that coheres with human judgments of contextual semantic similarity (Giulianelli et al., 2020). We probe BERT with synthetic inputs constructed by replacing each complex word with its space-delimited bases. Formally, given a sentence \( S = (w_1, \ldots, w_{i-1}, x, w_{i+1}, \ldots, w_n) \) where \( x \) denotes a blend or compound with contributing bases \( b_1, b_2 \), we record the average vector across \( x \)’s wordpiece tokens for each layer output in BERT’s transformer stack, \( e(l)(x) \), \( l \in 0, \ldots, 12 \), and compute its cosine similarity with the averaged vectors \( \frac{1}{2}(e(l)(b_1) + e(l)(b_2)) \) in the sequence \( S' = (w_1, \ldots, w_{i-1}, b_1, b_2, w_{i+1}, \ldots, w_n) \).
Figure 1: Pretrained BERT’s layer-wise similarity between representations of (a) complex OOVs and their base components; and (b) linear blends and “smoothies” (§3.2), lexical compounds forced to lose characters while remaining linear. All representations are computed using the original context in which the words appear. Error bars represent standard error of the mean over the class.

Figure 1(a) compares the per-layer similarities for blends with the two types of compounds described in §2: We find a clear distinction between blends and both compound classes. For compounds, BERT induces representations that are very similar to those of the components at all layers of the model. For blends, these representations diverge greatly, especially in the lower layers of the model, which capture surface-form characteristics of the input (Jawahar et al., 2019).

Since the difference between classes exists across all layers, we first wish to perform a more thorough analysis of possible reasons for it.

3.1 Semantics

One possible explanation for the difference in BERT’s treatment of blends and lexical compounds is that blends arise in lexical situations that are qualitatively different from those in which compounds are formed. This would lead to a different distribution of semantic relationships between bases of blends and compounds. In our annotated dataset we were able to witness such differences; for example, the ATTRIBUTE relation accounts for 23% of compounds but 38% of blends.

If BERT’s divergent treatment of blends and compounds is explained by the distribution over semantic categories for each complex word type, then we would expect the similarity scores within categories to be identical. Repeating the contextual similarity analysis within each semantic category, we find that there are substantial divergences between blends and compounds in several of the semantic categories. Figure 2 presents the similarity scores for the six relations containing at least 15 observations; blend representations are less similar to their decomposed versions compared to compounds regardless of the relation. A linear model trained to predict similarity confirms that blends are less similar to their components than compounds ($\rho = -.128, p < .001$).

3.2 Form

Another potential explanation is that differences in BERT’s treatment of blends and lexical compounds are driven by the form of each compound, rather than the meaning. On this view, the choice of whether to create a compound or a blend is a stylistic one (Renner, 2015), and so controlling for the character loss incurred in blends would produce the same processing difficulty for compounds.

Smoothies. If differences in surface form are what drives differences in contextualized representations, then transforming the compounds into mock-blends, which we term “smoothies”, should eliminate the differences between the two complex word types: we would expect the function of the similarity of a contextual encoding of a blend to its bases given a context it naturally occurs in to be approximately the same function of simi-
larity of a contextual encoding of a smoothie to its bases given the context the original compound occurs in. We create our smoothies using COPY-CAT (Kulkarni and Wang, 2018), a model which generates blends from two base forms via a sequence of character copy and delete actions learned over features extracted from an language model, an LSTM, and length-based heuristics. We train an ensemble of 50 COPY-CAT models on the blends from Deri and Knight (2015b) and apply them to our novel compounds. Since COPY-CAT can produce only linear blends, we compare the BERT correspondence for smoothies against linear blends only (whose aggregate similarities are notably similar to those of blends as a whole). In creating the smoothies, we made sure that the overall rate of lost characters (delete operations) is comparable to that of the true linear blends. We show in Figure 1(b) that smoothies pose similar generalization challenges as blends: the gap between linear blends and smoothies is small, while generalization for smoothies is far below that of the original compounds.

**Tokenization.** Having established that surface form is a main driver of the representational differences between blends and compounds, we now assess the specific impact of BERT’s tokenization model, WordPiece (WP). WordPiece is a trained model, consisting of a subword vocabulary constructed by identifying units (pieces) that appear repeatedly in a corpus. It distinguishes between word-initial pieces, which may be whole words, and word-noninitial pieces which are marked by a special “##” prefix. A word is then assigned a sequence of pieces whose characters matches it when concatenated. For example, WP(“segmenting”)=[‘segment’, ‘##ing’]. Such a model might be poorly suited to novel blends, which by definition reuse characters across bases, and which cannot be analyzed by traditional patterns of morphology.

To test the effect of segmentation, we provide WP with base-congruent segmentation points informed by their PAXOBS tags: for example, *shop-tics* is fed to BERT as → sh+##op+##tics. We find that this change does little to bridge the gap between blends and compounds: a redrawn version of Figure 1(a) using this tokenization is almost identical to the original. Upon further examination, we find that while pre-tokenizing with PAXOBS results in a larger number of wordpiece tokens (an average of 4.55 vs. 3.30), a similar leap occurs in compounds (3.41 vs. 2.48), suggesting that WP does not produce morphologically accurate segments for compounds either (Bostrom and Durrett, 2020). The crux of the issue must therefore lie within BERT.

In conclusion, we have shown that the root cause of blend mistreatment in large contextual transformer models is their form, although knowing only their sequence structure is not sufficient. Therefore, in the following section we suggest models which attempt to identify blend segmentation points, but also ones which attempt recovery of their original bases, in order to place them in an appropriate topical context.

### 4 Will it Unblend?

We next test to what extent existing models can help systems understand the meaning of novel blends, an aspect of human language understanding that has been little explored in NLP evaluation tasks. As demonstrated in §3, successfully
representing blends requires the capability to both properly decompose their form and identify the original constituents. We therefore cast blend understanding as two tasks: segmenting blends into character sequences (§4.1), and recovering blends’ bases post-segmentation (§4.2). We leave the task of recognizing blends to future work.\footnote{Note that the best OOV classification baseline in Pinter et al. (2020), a ridge classifier using character n-gram features, reaches \(.305\) F1 on the blends class (its macro-F1 on all 18 classes is \(.323\)).}

**Compounds.** Compounds were used as a comparative class in §3, but for the purpose of form understanding we focus on blends. For compounds with a known segmentation, base recovery is trivial, as each side of the segmentation point is always a base. As for segmentation, we have shown in §3.2 that BERT’s transformer layers are capable of recovering from poor WordPiece performance, and so the utility of segmenting compounds is limited compared to blends.\footnote{We nevertheless evaluated the segmenters on compounds (compare with Table 2). WP performs about as well as on blends in F1 (\(.558\)), and better in exact match (\(.34\%\)), and Domain Unigram LM outperforms it on both (\(.630\), \(.39\%\) respectively). In compounds, lenient and strict metrics converge.} Knowing that words are kept in their original form can define much simpler and more effective systems of discovery than the ones described below, such as a dictionary lookup of both sides for each possible single segmentation point.

### 4.1 Blend Segmentation

We approximate the problem of inferring blend structure by defining a segmentation task over the character sequence which is the blend form, on the rationale that supplying a downstream system with character segments, each coherently representing a single known word or morpheme, would improve its ability to represent the input sequence. For example, a character-aware system familiar with non-complex words might understand that the initial \(h\)at from \(h\)atriot\(i\)sm is related to \(h\)ate if given in isolation; but with \(h\)atr it would be at a loss.

**Metrics.** We draw on our PAXOBS schema (§2) to define segment-level precision and recall scores for a given blend (e.g. \(s\)hoot\(p\)tics: \(A\)XXXB\(B\)B\(B\)S). A system’s prediction is a set of character indices where segmentation should occur. We count any index which separates characters of the same label as a false positive, towards precision (e.g. the seg-

| Model                | #segs | Prec. L F1 | S F1 | L EM |
|----------------------|-------|-----------|------|------|
| All-chars            | 10.15 | \(.272\)  | \(.427\) | \(.427\) | 0%    |
| Seq. tagger          | 4.70  | \(.291\)  | \(.400\) | \(.376\) | 5%    |
| WordPiece            | 3.30  | \textbf{.450} | \textbf{.562} | \(.484\) | 22%   |
| Domain BPE           | 3.18  | \(.408\)  | \(.517\) | \(.441\) | \textbf{24%} |
| Domain ULM           | 4.08  | \(.428\)  | \(.556\) | \textbf{.492} | 22%   |

Table 2: Results for segmentation \((N = 142, \text{macro-aggregation})\). “L” – lenient, “S” – strict, “EM” – exact match. The desired number of segments varies between 2.05 and 3.40, depending on affixes and recall policy.

**Systems.** We compare the following systems (see Appendix A.2 for implementation details):

- **All-chars.** A baseline which marks every character as its own segment (perfect recall).
- **Sequence Tagger.** We annotated the 1,579 blends in Gangal et al. (2017)’s dataset\footnote{After filtering duplicates from its 1,624 lines.} for PAXOBS tags and used them for training a supervised neural character-level tagger, whose results are converted into segmentations.
- **WordPiece.** We run WP “out of the box”.
- **In-domain Subwords.** We train BPE (Sennrich et al., 2016) and Unigram LM (Kudo, 2018) subword tokenizers on news data from the Corpus of Contemporary American English (1990–2015; Davies, 2008) using the sentencepiece package (Kudo and Richardson, 2018), set to the same vocabulary size as the WP model.

**Results.** The results in Table 2 show that all models struggle to find correct segmentation, even compared to the all-chars baseline. The low performance of the supervised tagger suggests that little can be inferred from relative character placement, demonstrating the highly variable nature of novel blends. Corpus-based segmentation models manage to segment over 20% of the
blends successfully. This number is slightly higher when looking at the subset of linear blends: Domain BPE matches 29% of them exactly. Further analysis of the WP segmentations reveals a weakness in cases where the first post-A character suggests a plausible continuation to base A, common enough to appear in WP’s vocabulary, e.g. [males:stream] (true bases male, mainstream; labels XXAABBAAAA, or [chip:ster] (true bases chico, hipster; labels AXBAAAAA).

We next consider the challenge of reconstructing the base components for segmented blends.

4.2 Blend Component Recovery

We tasked different models with identifying the contributing bases \((A, B)\) out of all possible words given a gold-segmented blend and an input vocabulary. We create sets of candidate words for each blend which could, in principle, create the same blend as the true bases. For example, the blend \textit{trash} = \textit{three} + \textit{couple} will induce candidates such as thrash for \(A\) and example for \(B\). We report the following metrics:

- **MRR**(\(A, B, \omega\)) is the mean reciprocal rank of true base \(A/B\) across all possible candidates for that side (\textit{single-side prediction}), or \(\omega\) of the true base pair out of all possible base pair candidates (\textit{pair prediction});

- **Precision @1** is the proportion of blends for which the top candidate is the true base pair.\(^{18}\)

In order to maintain a fair comparison between the models (see below), we extracted the candidate lists for all model evaluations from the GloVe model’s vocabulary, as it is the only one restricted for in-vocabulary testing. In total, 33 of the candidate lists (12%) are singletons, including two blends where neither base has negative samples. Six blends (4%) lacked the correct base for one of the sides, and these cases were treated as ranked last among candidates; three of these lists were empty, translating to a #1 rank for all systems. The lower bounds on the metrics resulting from these candidate list limitations are presented at the top of Table 3.\(^{20}\)

**BERT RANKER.** We propose a contextual representational approach for ranking two-sided base candidates using iterative piece prediction: we replace each appearance of a blend \(b\) in its context sentence \((w_1, \ldots, w_{i-1}, \hat{b}, w_{i+1}, \ldots, w_n)\) with two successive [MASK] tokens: \((w_1, \ldots, w_{i-1}, m_1, m_2, w_{i+1}, \ldots, w_n)\). Then, we use a pretrained BERT masked language model to compute wordpiece prediction distributions for these masked tokens. We sort all possible candidate base pairs \((l, r)\) according to the sum of probabilities for their bases’ first pieces, \(P(m_1 = l_0) + P(m_2 = r_0)\),\(^{21}\) and record the rank of the true base pair. When candidate pairs have the same first-piece pair, we break ties by iteratively predicting the next pieces after inserting the shared pieces into the input. We also implement an ablation (\(-\text{CONTEXT}\)) where no context is added to the masks, in order to evaluate the contribution of the sentence contexts in isolation.

For single-side metrics, we report the rank of the true base in the prediction distribution of a single [MASK] token (instead of two); in another variant (\(+\text{-OTHER-BASE}\)) we add the true base from the other side to the context, in order to level the playing field with the baselines, which we describe next (see Appendix A.3 for implementation details):

- **Character RNN.** We separately train a forward and a backward character-level RNN on over 100,000 documents from the Westbury corpus (Shaoul, 2010). We feed the blend’s left (right) context to the forward (backward) RNN, then record the probability of each \(A\) (\(B\)) candidate as a continuation of the context, computed as the average of character log-likelihoods.

- **Edit distance (ED).** Following Cook and Stevenson (2010), we compare the string similarity (Levenshtein distance) between base candidate pairs’ orthographic forms.\(^{22}\) Single-side

---

\(^{18}\)In three-based blends, we denote the last base as \(B\) for this task, surmising it is more important than the second base given the right-headedness tendency of English.

\(^{19}\)This contrasts with precision as reported by Cook and Stevenson (2010), who count a pair as correct if at least one base is correct.

\(^{20}\)The full lists are available on the repository.

\(^{21}\)This is crucial, since predicting a ‘#’-initial suffix token effectively attaches it to the preceding token.

\(^{22}\)A variant using phonological forms, extracted from a phonological lexicon (Lee et al., 2020), was limited by only having pronunciations for a fraction of bases and candidates. In cases where both base pronunciations were found the ranking was good, hinting at a promising avenue for future work by implementing automatic text-to-phone modeling.
Table 3: Results for component recovery. *Results dependent on knowledge of the correct base on the other side.

| Model                  | MRR- $A$ | MRR- $B$ | MRR- $\omega$ | P@1 |
|------------------------|----------|----------|----------------|-----|
| Lower bound            | .115     | .257     | .036           | .014 |
| Character RNN          | .162     | .368     | .060           | .021 |
| Edit distance          | .176*    | .432*    | .066           | .014 |
| fastText               | .357*    | .610*    | .167           | .127 |
| GloVe                  | .449*    | .734*    | .188           | .127 |
| BERT RANKER + OTHER-BASE | .392    | .711     | .288           | .264 |
| BERT RANKER − CONTEXT  | .379     | .675     | .147           | .127 |
| BERT RANKER + OTHER-BASE | .379    | .668     | .137           | .127 |

Results. Results are presented in Table 3. We note the higher performance on $B$ bases achieved by all models, a fact which advantages WordPiece which leaves word-initial pieces unmarked (see §3.2), as opposed to models such as XLM (Conneau et al., 2019), which mark word-final pieces. If the beginning of the blend bears more resemblance to the base it originated from, there’s a better chance of properly representing that base in the overall blend. These findings suggest an iterative setup, where first the $B$ base is predicted and only then $A$ is matched, might prove more successful. We leave this variant to future work.

Our BERT RANKER model outperforms all baselines in the more realistic full-word setting (MRR-$\omega$, P@1). When ranking single bases, it does not benefit much from awareness of the true other base (the oscillations recorded in the table are too small to be meaningful), suggesting that most of its power lies in processing context and not in word form representation. This conclusion is further supported by the superior performance of the static type-level GloVe embeddings, whose lead over fastText and BERT−CONTEXT in all MRR measures suggests that word form is less helpful even in uncontextualized settings. The particularly poor performance of the character RNN and edit distance model shows that it is difficult to learn the task without any semantic signal.

Error Analysis. A qualitative assessment of the contexts which help BERT RANKER to predict bases perfectly relative to the −CONTEXT variant shows that they typically contain one or more of the bases in their entirety (e.g., eggerssories appears near multiple occurrences of the word eggs). By contrast, in some longer contexts containing diverse topics, the inclusion of context wipes out the accessibility of the component bases, typically the first one (e.g. chesticle, in which the context does not mention body parts, or cancerchondria which mentions the word condition but neither of the bases).

5 Related Work

Prior work on blends has largely focused on generation (e.g., Das and Ghosh, 2017; Simon, 2018; Deri and Knight, 2015b; Kulkarni and Wang, 2018; Smith et al., 2014). While Gangal et al. (2017) provide a unified dataset of 1,579 blends, annotated for bases, they do not provide contexts for real-world appearances of the blends, nor a breakdown of the semantic relationship between their constituents. Moreover, some are synthetically generated by a seq2seq model. In addition, these works all restrict their models to linear two-word blends. Our PAXOBS scheme handles non-linear and multi-base blends.

Cook and Stevenson (2010) presented a non-contextual method for blend base detection using a dictionary-based lexicon, evaluated over an unreleased dataset, and Ek (2018) used features from static embeddings to unblend words in Swedish. We adopt the candidate-ranking approach of these works to evaluate component recovery, but incorporate context with context-sensitive language models, and add the task of blend segmentation.

Extracting the semantics of constituents from larger phrases is not a problem unique to single-token blends. Shwartz and Waterson (2018) worked on multi-word compounds; Maddela et al.
(2019) segment hashtags, roughly half of which are akin to our notion of compounds, by training a neural scoring system over features extracted from word form, dictionary lookup and language model probabilities. Another connection is to the learning of morphological rules, e.g., for processes such as derivation (Kondratyuk, 2019) and lemmatization (Chrupala, 2006; Ullman et al., 1976; Hirschberg, 1977). Cotterell and Schütze (2018) present a supervised model of derivational morphology that jointly accounts for segmentation as well as composition of static word embeddings from the embeddings of morphemes, thereby touching on two of the main tasks undertaken in our paper. However, the application of such a model to blends is complicated by the relative lack of labeled training data, as well as the irregularity of the underlying phenomenon.

Novel blends are an example of linguistic creativity, which frequently operates at the subword level. Related phenomena include eggcorns, which are alternative spellings that yield an apparently more transparent relationship between form and function (Reddy, 2009); puns, which substitute words in new contexts based on phonological similarity (Jaech et al., 2016); respellings that attempt to reintroduce prosodic expression into spelling (Brody and Diakopoulos, 2011); intentional obfuscation (Zalmout et al., 2019); and typographical errors (Heigold et al., 2017). We therefore view blends as an instance of a broad set of creative phenomena that poses challenges for the token-based approaches that currently dominate natural language processing.

6 Conclusion

This work focuses on the challenge of interpreting novel blends, which requires integrating subword structure and contextual features. We present a new dataset annotated using a novel character-level schema as well as for semantic tags, and offer preliminary evaluations showing that (a) blends are handled differently than compounds by BERT, due mostly to the phenomenon of character loss; (b) existing tokenizers generally do not respect blend boundaries; and (c) recovering the components of a blend is a difficult task which challenges word-form, distributional, and contextual approaches. Our results further highlight that annotation schemata such as those of Tratz and Hovy (2010), which were designed for noun compounds, are generalizable to other relational word types. In future work, we plan to integrate these signals into a better blend processor, and to further address the effect of blends on downstream tasks from semantic and syntactic viewpoints. In addition, we aim to further examine the methods from our experiments on other classes of novel words and in other languages. We also plan to add phonetic resources for improving treatment of non-linear blends.

Acknowledgments

We thank Max Bittker for providing us with details and logs from the NYT bot. We thank Lelia Glass, Kyle Gorman, Arya McCarthy, Ian Stewart, Kristina Toutanova, Sarah Wiegreffe, Yuwei Wu, Wei Xu, Dyi Yang, and the anonymous reviewers for their valuable notes. YP is a Bloomberg Data Science PhD Fellow. CJ is a postdoctoral research associate supported by NSF grant 1849236 awarded to Maryellen MacDonald.

References

John Algeo. 1977. Blends, a structural and systemic view. American speech, 52(1/2):47–64.
Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. arXiv preprint arXiv:2004.03720.
Thorsten Brants. 2000. Tnt: a statistical part-of-speech tagger. In Proceedings of the sixth conference on Applied natural language processing, pages 224–231. Association for Computational Linguistics.
Samuel Brody and Nicholas Diakopoulos. 2011. Cooooooooooooooollllllllllll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs. In Proceedings of the conference on empirical methods in natural language processing, pages 562–570. Association for Computational Linguistics.
Grzegorz Chrupala. 2006. Simple data-driven context-sensitive lemmatization. Procesamiento del lenguaje natural, (37):121–130.
Grzegorz Chrupala, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morfette. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08), Marrakech, Morocco. European Language Resources Association (ELRA).
Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Gymzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.
Paul Cook and Suzanne Stevenson. 2010. Automatically identifying the source words of lexical blends in English. *Computational Linguistics*, 36(1):129–149.

Ryan Cotterell and Hinrich Schütze. 2018. Joint semantic synthesis and morphological analysis of the derived word. *Transactions of the Association for Computational Linguistics*, 6:33–48.

Kollol Das and Shaona Ghosh. 2017. Neuramanteau: A neural network ensemble model for lexical blends. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 576–583.

M. Davies. 2008. The corpus of contemporary american english: 450 million words, 1990-present.

Aliya Deri and Kevin Knight. 2015a. How to make a frenemy: Multitape FSTs for portmanteau generation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 206–210.

Aliya Deri and Kevin Knight. 2015b. How to make a frenemy: Multitape FSTs for portmanteau generation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 206–210, Denver, Colorado. Association for Computational Linguistics.

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.

Corina Dima and Erhard Hinrichs. 2015. Automatic noun compound interpretation using deep neural networks and word embeddings. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 173–183, London, UK. Association for Computational Linguistics.

Adam Ek. 2018. Identifying source words of lexical blends in swedish. *Thanks to our sponsors: Gold: Stora Skuggans Vårdshus Silver: TT Nyhetsbyran and Lingsoft Bronze: Convertus, Digital Grammars, IQVIA and Voice Provider*, page 82.

Varun Gangal, Harsh Jhamtani, Graham Neubig, Edward Hovy, and Eric Nyberg. 2017. Charmanteau: Character embedding models for portmanteau creation. *arXiv preprint arXiv:1707.01176*.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Seattle, USA. Association for Computational Linguistics.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2017. How robust are character-based word embeddings in tagging and mt against word scrambling or random nouse? *arXiv preprint arXiv:1704.04441*.

Daniel S. Hirschberg. 1977. *Algorithms for the longest common subsequence problem*. *J. ACM*, 24(4):664–675.

Aaron Jaech, Rik Koncel-Kedziorski, and Mari Osterdorff. 2016. Phonological pun-derstanding. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 654–663.

Ganesh Jawahar, Benoit Sagot, and Djamel Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dan Kondratyuk. 2019. Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–18, Florence, Italy. 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.

Vivek Kulkarni and William Yang Wang. 2018. Simple models for word formation in slang. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1424–1434, New Orleans, Louisiana. Association for Computational Linguistics.
A “Will it Unblend?”: Supplementary

A.1 Annotation
When annotating the PAXOBS schema, the A base is defined as the one whose exclusive material precedes all other exclusive materials in the blend, and from then on iteratively through the alphabet.

A.2 Segmentation Experiment

Character Tagger. We manually annotated 550 of the 1,579 blends in the (Gangal et al., 2017) dataset, and passed the rest through a heuristic program to cover whatever linear blends remained. The program flagged 150 blends as suspected non-linear, so we manually annotated them as well. A tagger trained on only the 550-blend set originally annotated did not reach better F1 scores than the one reported in the main text, trained on the full set.

We use a 3-layer Bidirectional LSTM followed by a 4-layer MLP (tuned in the 2–4 range on dev) with ReLU activation (tuned vs. tanh), trained for 30 epochs with early stopping, optimized using Adam (Kingma and Ba, 2014) with learning rate 0.01 and a batch size of 96 (not tuned). The character embedding dimension is 200 and the hidden dimensions for the LSTM and MLP are both 192 (not tuned). Tagging accuracy on the dev set is \(457\) and on our dataset \(462\). We translate the resulting PAXOBS tags into segmentations by segmenting on each label change (so \(\langle \text{shoptics}, AAXXBBBS \rangle\) becomes “sh; op; tics”). The model is implemented in PyTorch (Paszke et al., 2019).

Domain models. The vocabulary size of BPE and the Unigram LM are by default automatically inferred, but may be specified. For BPE, we tested several vocabulary sizes: 20,000, 30,000, 30,522 (WP vocabulary size) and 40,000, selected heuristically, and 30,522 performed best. Consequently, we used the same size for Unigram LM.

Results. Strict exact match scores are: All-chars, 0.0%; Seq. tagger, 3.5%; WordPiece, 7.7%; Domain BPE, 10.6%; Domain ULM, 6.3%. This ordering is consistent with the lenient exact match scores reported in Table 2.

A.3 Recovery Experiment Details
A single hyperparameter controlling the minimum length of candidate overlap in linear blends was set to 3 with no tuning. Candidates were selected ac-

More text to be added.