Lexical Semantic Recognition

Nelson F. Liu
Stanford University
nfliu@cs.stanford.edu

Daniel Hershcovich
University of Copenhagen
dh@di.ku.dk

Michael Kranzlein **Nathan Schneider**
Georgetown University
{mmk119, nathan.schneider}@georgetown.edu

Abstract

In lexical semantics, full-sentence segmentation and segment labeling of various phenomena are generally treated separately, despite their interdependence. We hypothesize that a unified *lexical semantic recognition* task is an effective way to encapsulate previously disparate styles of annotation, including multiword expression identification/ classification and supersense tagging. Using the STREUSLE corpus, we train a neural CRF sequence tagger and evaluate its performance along various axes of annotation. As the label set generalizes that of previous tasks (PARSEME, DiMSUM), we additionally evaluate how well the model generalizes to those test sets, finding that it approaches or surpasses existing models despite training only on STREUSLE. Our work also establishes baseline models and evaluation metrics for integrated and accurate modeling of lexical semantics, facilitating future work in this area.

1 Introduction

Many NLP tasks traditionally approached as tagging focus on lexical semantic behavior—they aim to identify and categorize lexical semantic units in running text using a general set of labels. Two examples are supersense tagging of nouns and verbs as formulated by Ciaramita and Altun (2006), and verbal multiword expression (MWE) identification and classification in the multilingual PARSEME shared tasks (Savary et al., 2017; Ramisch et al., 2018, 2020). By analogy with named entity recognition, we can use the term **lexical semantic recognition** (LSR) for such chunking-and-labeling tasks that apply to lexical meaning generally, not just entities. This disambiguation can serve as a foundational layer of analysis for downstream applications in natural language processing, and provides an initial level of organization for compiling lexical resources, such as semantic nets and thesauri.

In this paper, we tackle a more inclusive LSR task of lexical semantic segmentation and disambiguation. The STREUSLE corpus (see §2) contains comprehensive annotations of MWEs (along with their holistic syntactic status) and noun, verb, and preposition/possessive supersenses. We train a neural CRF tagger (Lafferty et al., 2001) using BERT embeddings (Devlin et al., 2019) and find that it obtains strong results as a first baseline for this task in its full form.

In addition, we ask: Does a tagger trained on STREUSLE generalize to evaluations like the PARSEME shared task on verbal MWEs (Ramisch et al., 2018) and the DiMSUM shared task on MWEs and noun/verb supersenses (Schneider et al., 2016)? Results show our LSR model based on STREUSLE is general enough to capture different types of analysis consistently, and suggest an integrated full-sentence tagging framework is valuable for explicit modeling of lexical semantics in NLP.[1]

2 LSR Tagging Frameworks

Our tagger is based on STREUSLE (Supersense-Tagged Repository of English with a Unified Semantics for Lexical Expressions; Schneider and Smith, 2015; Schneider et al., 2018),[2] a corpus of web reviews annotated comprehensively for lexical semantic units and supersense labels. Specifically, there are three annotation layers: **multiword expressions**, **lexical categories**, and **supersenses**. The supersenses apply to noun, verb, and prepositional/possessive units. Figure 1 shows an example.

Many of the component annotations have been applied to other languages: verbal multiword expressions (Savary et al., 2017; Ramisch et al., 2018), noun and verb supersenses (e.g., Picca et al.,

[1]Code, pretrained models, and model and scorer output (all train/dev/test splits) can be found at https://nelsonliu.me/papers/lexical-semantic-recognition

[2]https://github.com/nert-nlp/streusle

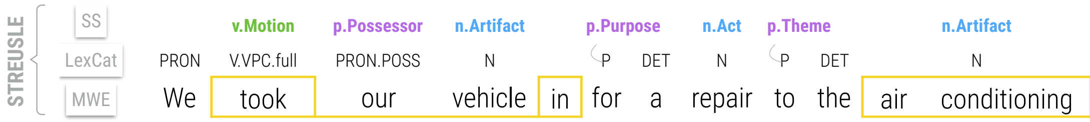

Figure 1: Example annotated sentence from the STREUSLE training set. The (strong) multiword expressions "took…in" and "air conditioning" each receive a single lexcat and supersense. UD syntax is not shown.

2008; Qiu et al., 2011; Schneider et al., 2013; Martínez Alonso et al., 2015; Hellwig, 2017), and adposition supersenses (Hwang et al., 2017; Zhu et al., 2019). In this paper we focus on English, where comprehensive annotation is available.

2.1 STREUSLE Annotation Layers

STREUSLE comprises the entire 55K-word Reviews section of the English Web Treebank (Bies et al., 2012), for which there are gold Universal Dependencies (UD; Nivre et al., 2020) graphs, and adopts the same train/dev/test split.

The lexical-level annotations do not make use of the UD parse directly, but there are constraints on compatibility between lexical categories and UPOS tags (see §3).

Multiword expressions (MWEs; Baldwin and Kim, 2010) are expressed as groupings of two or more tokens into idiomatic or collocational units. As detailed by Schneider et al. (2014a,b), these units may be contiguous or *gappy* (discontinuous).[3] Each unit is marked with a binary *strength* value: idiomatic/noncompositional expressions are *strong*; collocations that are nevertheless semantically compositional, like "highly recommended", are *weak*.

We use the term **lexical unit** for any expression that is either a *strong* MWE grouping of multiple tokens, or a token that does not belong to a strong MWE. Every token in the sentence thus belongs to exactly one lexical unit. The other layers of semantic annotation augment lexical units, and weak MWEs are groupings of (entire) lexical units.

Lexical categories (lexcats) describe the syntax of lexical units. They are similar to UPOS tags available in the UD annotations of the corpus, but are necessary in order to (a) express refinements relevant to the criteria for the application of supersenses, and (b) account for the overall syntactic behavior of strong MWEs, which may not be obvious from their internal syntactic structure.[4] Appendix A gives the full list of lexcats.

[3]The gap in a discontinuous MWE may contain single-word and/or other multiword expressions, provided that those embedded MWEs do not themselves contain gaps.

[4]This is also done in other resources (e.g., Shigeto et al., 2013; Gerdes et al., 2018).

Supersenses semantically classify lexical units and provide a measure of disambiguation in context. There are 3 sets of supersense labels: nominal, verbal, and prepositional/possessive. The lexcat determines which of these sets (if any) should apply.[5]

The MWE, lexcat, and supersense information over lexical units is serialized as per-token tags in a BIO-based encoding (details in §2.1.1).

2.1.1 Tag Serialization

STREUSLE specifies **token-level tags** to allow modeling lexical semantic recognition as sequence tagging. The `BbIiOo_~` tagging scheme (Schneider et al., 2014a) consists of 8 positional flags indicating MWE status: `O` applies to single-word expressions, `B` to the start of a new MWE, `I_` to the continuation of a strong MWE, and `I~` to the continuation of a weak MWE (if not continuing a strong MWE within the weak MWE). The lowercase counterparts `o`, `b`, `i_`, `i~` are the same except they are used within the gap of a discontinuous MWE. For MWE identification, local constraints on tag bigrams—e.g., that the bigrams ⟨`B`,`B`⟩ and ⟨`B`,`O`⟩ are invalid, and that the sentence must end with `I_`, `I~`, or `O`—ensure a valid overall segmentation into units (Schneider and Smith, 2015).

The lexcat and (where applicable) supersense information is incorporated in the *first* tag of each lexical unit.[6] Thus `B-N-n.ARTIFACT` indicates the

[5]Some preposition units are labeled with two supersenses drawn from the same label set: the **scene role** label represents the semantic role of the prepositional *phrase* marked by the preposition, and **function** label represents the lexical contribution of the *preposition* in itself (Schneider et al., 2018). The scene role and the function are identical by default.

[6]Though in named entity recognition it is typical to include the class label on every token in the multiword unit, STREUSLE does not do this because it would create a non-local constraint across gaps (that the tags at either end have matching lexcat and supersense information). A tagger would either need to use a more expensive decoding algorithm or would need to greatly enhance the state space so within-gap tags capture information about the gappy expression.

In STREUSLE there is actually a slight limitation due to the verbal lexcats, which distinguish between single-word and strong multiword expressions (see Appendix A): if a `B-*` or `I~-*` tag is followed by a gap, there is no local indication of whether the expression will be strong or weak (strength is indicated only after the gap). If the expression being started is strong, then one of the verbal MWE subtypes (`V.VID`, etc.)

beginning of an MWE whose lexcat is `N` and supersense is N.ARTIFACT. `I_` and `i_` tags never contain lexcat or supersense information as they continue a lexical unit, whereas `O`, `B`, `I~`, `o`, `b`, and `i~` always do. Figure 2 illustrates the full tagging. All told, STREUSLE has 601 complete tags.

```
We/O-PRON          took/B-V.VPC.full-v.Motion
our/o-PRON.POSS  vehicle/o-N-n.ARTIFACT  in/I_
for/O-P-p.Purpose   a/O-DET   repair/O-N-n.ACT
to/O-P-p.Theme   the/O-DET   air/B-N-n.ARTIFACT
conditioning/I_
```

Figure 2: Serialization as token-level tags for the example sentence from figure 1.

2.2 Related Frameworks

The Universal Semantic Tagset takes a similar approach (Bjerva et al., 2016; Abzianidze and Bos, 2017; Abdou et al., 2018), and defines a cross-linguistic inventory of semantic classes for content and function words, which is designed as a substrate for compositional semantics, and does not have a trivial mapping to STREUSLE categories.

However, two shared task datasets consist of subsets of the categories used for STREUSLE annotations, on text from different sources.

PARSEME Verbal MWEs. The first such dataset is the English test set for the PARSEME 1.1 Shared Task (Ramisch et al., 2018), which covers several genres (including literature and several web genres) and is annotated only for verbal multiword expressions. The STREUSLE lexcats for verbal MWEs are identical to those of PARSEME; thus, a tagger that predicts full STREUSLE-style annotations can be evaluated for verbal MWE identification and subtyping by simply discarding the supersenses and the non-verbal MWEs and lexcats from the output.

DiMSUM. The second shared task dataset is DiMSUM (Schneider et al., 2016), which was annotated in three genres—TrustPilot web reviews, TED talk transcripts, and tweets—echoing the annotation style of STREUSLE when it contained only MWEs and noun and verb supersenses. DiMSUM does not contain prepositional/possessive supersenses or lexcats. It also lacks weak MWEs.

3 Modeling

We develop and evaluate a strong neural sequence tagger on the full task of lexical semantic recognition with MWEs and noun/verb/preposition/possessive supersenses to assess the performance of modern techniques on the full joint tagging task. Our tagger feeds pre-trained BERT representations (Devlin et al., 2019) through a biLSTM. An affine transformation followed by a linear chain conditional random field produces the final output. For further implementation details, see Appendix B.

The predicted tag for each token is the conjunction of its MWE, lexcat, and supersense.[7] There are 572 such tags in the STREUSLE training set, and only 12 unique conjoined tags in the development set are unseen during training (≈5% of the development set tagging space, corresponding to ≈0.2% of the tokens in the development set).

Constrained Decoding. A few hard constraints are imposed in tagging. To enforce valid *MWE chunks*, we use first-order Viterbi decoding with the appropriate corpus-specific constraints (e.g., for STREUSLE MWEs, the `BbIiOo_~` tagset; see §2.1.1). The MWE constraint is applied during training and evaluation. In addition, a given token's possible lexcats are constrained by the token's *POS tag and lemma*. For instance, a token with the `AUX` UPOS tag can only take the `AUX` lexcat. However, if the token's UPOS is `AUX` and its lemma is "`be`", it can take either the `AUX` or `V` lexcats.

The POS and lemma constraints are only applied during evaluation; to avoid relying on gold POS/lemma annotations at test time we use an off-the-shelf system (Qi et al., 2018).

3.1 Experiments

We train the tagger on version 4.3 of the English STREUSLE corpus and evaluate on the STREUSLE, English PARSEME, and DiMSUM test sets (§2). The latter two are (zero-shot) out-of-domain test sets; the tagger is not retrained on the associated shared task training data.

We also compare to a model with static word representations by replacing BERT with the concatenation of 300-dimensional pretrained GloVe embeddings (Pennington et al., 2014) and the output of a character-level convolutional neural net-

should apply; whereas the correct lexcat for a single-word verb is plain `V`. In practice this is not a problem.

[7] For prepositions and possessives, the supersense is either a pair of labels, or a single label serving dually as scene role and function (fn. 5).

STREUSLE 4.3 (test, 5,381 words)	Tags Full	−LC	−SS	NOUN Labeled F	VERB Labeled F	SNACS Labeled F	Role F	Fxn F	MWE LinkAvg P	R	F	VERB MWE ID F
	Accuracy											
# Gold	5381			986	697	485			433.5			66
BERT GloVe (Gold)	82.5 79.3	82.7	89.9	69.0 66.1	77.1 72.1	71.4 61.0	72.4	81.7	80.0	64.9	71.6 59.5	63.9 38.6
BERT GloVe (Pred.)	81.0 77.5	81.7	87.9	68.0 65.7	75.1 70.0	71.6 58.0	72.4	82.8	77.6	63.1	69.5 60.3	62.3 43.0
BERT GloVe (None)	82.0 77.1	82.7	89.1	69.6 64.9	76.8 70.3	70.9 58.1	71.9	81.0	82.0	64.3	72.0 60.3	63.9 42.5
Schneider et al.	–	–	–	–	–	55.7	58.2	66.7	–	–	–	–

Table 1: STREUSLE test set results (%). (Gold): gold POS/lemmas (used in constraints only). (Pred.): predicted POS/lemmas. (None): MWE constraints only. −LC: excluding lexical category. −SS: excluding supersense. Labeled F: labeled identification F_1-score. SNACS: preposition supersenses. MWE LinkAvg P, R, F: evaluates MWE identification with partial credit. Identification of verbal MWEs (exact match) is equivalent to the PARSEME MWE-based metric. Schneider et al. (2018): previous best full SNACS tagger, reported on STREUSLE 4.0.

PARSEME 1.1 (EN-test, 71,002 words) MWE-based P	R	F	Token-based P	R	F		DiMSUM 1.0 (test, 16,500 words) MWEs P	R	F	Supersenses P	R	F	Combined Acc	P	R	F
501			1087			**# Gold**	1115			4745			5860			
36.1	45.5	40.3	40.2	52.0	45.4	BERT (Gold)	47.9	52.2	50.0	52.1	56.5	54.2	76.9	51.3	55.7	53.4
34.1	45.9	39.2	37.1	52.2	43.4	BERT (Pred.)	48.8	50.7	49.7	49.1	53.9	51.4	75.1	49.1	53.3	51.1
36.2	45.3	40.3	40.4	51.8	45.4	BERT (None)	53.0	49.2	51.0	50.8	55.1	52.9	76.5	51.2	53.9	52.5
33.8	32.7	33.3	37.3	31.8	34.4	Nerima+ Kirilin+	73.5	48.4	58.4	56.8	59.2	58.0	85.3	59.0	57.2	58.1
–	–	36.0	–	–	40.2	Taslimipoor+										
–	–	41.9	–	–	–	Rohanian+										

Table 2: PARSEME and DiMSUM zero-shot test set results (%) for BERT models from table 1, compared to prior published results on the tasks. GloVe F1 scores (not shown) are 17–20 points below the corresponding BERT scores for PARSEME, and 14–15 for DiMSUM. Kirilin et al. (2016): the best performing system from Schneider et al. (2016). Kirilin et al. (2016) and other shared task systems had access to gold POS/lemmas and Twitter training data in addition to all of STREUSLE for training. Nerima et al. (2017): a rule-based system which performed best for English in the shared task (Ramisch et al., 2018). Taslimipoor et al. (2019), Rohanian et al. (2019): more recent results on the test set (both used ELMo and dependency parses; only some scores were reported).

work. Finally, we also establish an upper bound on performance by providing the model with gold POS tags and lemmas; note that the difference between gold and predicted POS tags and lemmas only applies to the constrained decoding.

3.2 Results and Discussion

Table 1 shows all standard **STREUSLE** evaluation metrics on the test set. For preposition supersenses (SNACS), we compare to the results in Schneider et al. (2018), who performed MWE identification and supersense labeling for prepositions only. Note that Schneider et al. (2018) used version 4.0 of the STREUSLE corpus, which is slightly different from the version we use (some of the SNACS annotations have been revised). However, our baseline tagger, even with GloVe embeddings, outperforms Schneider et al. (2018) on that subset. Using BERT embeddings with constraints POS tags and lemmas improves performance substantially; on preposition supersense tagging, it even outperforms using gold POS tags and lemmas. Liu et al. (2019) also found that BERT embeddings improved SNACS labeling on STREUSLE 4.0, although they study a simplified setting (gold preposition identification, and only considering single words).

Table 2 shows standard **PARSEME** and **DiMSUM** test set evaluation metrics, for models trained on the STREUSLE training set, in a zero-shot out-of-domain evaluation setting. On the PARSEME test set, our BERT-based model approaches the state-of-the-art MWE-based F-score and exceeds the best reported *fully-supervised* token-based F-score. However, on the DiMSUM test set, the BERT model did not outperform the best shared task system, likely owing to the comparative difficulty of the full lexical semantic recognition task versus the restricted DiMSUM setting.

These results demonstrate that pre-training contextualized embeddings on large corpora can help models generalize to out-of-domain settings.[8]

Constrained decoding does not substantially impact the performance of our BERT model. In general, constraints with gold POS/lemmas perform the best, while not using POS/lemma constraints is

[8] A small fraction of sentences in the PARSEME test set (194/3965) are EWT reviews sentences that also appear in STREUSLE's dev set. The rest of the PARSEME test set contains other web and non-web genres (Walsh et al., 2018), and thus it is mostly out-of-domain relative to STREUSLE. None of the PARSEME training set overlaps with STREUSLE.

I	have	a	new	born	daughter	and	she	helped	me	with	a	lot
O-PRON	*O-V-v.stative*	*O-DET*	**O-ADJ**	*I_*	*O-N-n.PERSON*	*O-CCONJ*	*O-PRON*	*O-V-v.social*	**o-PRON**	*O-P-p.Theme*	*B-DET*	*I_*
O-PRON	*O-V-v.stative*	*O-DET*	*B-ADJ*	*I_*	*O-N-n.PERSON*	*O-CCONJ*	*O-PRON*	*O-V-v.social*	*O-PRON*	*O-P-p.Theme*	*B-DET*	*I_*

Go	down	1	block	to	Super	8	.
B-V.VPC.semi-v.motion	*I_*	*O-NUM*	*O-N-n.COGNITION*	*O-P-p.Goal*	**B-N-n.LOCATION**	*O-NUM*	*O-PUNCT*
O-V-v.motion	*O-P-p.Direction*	*O-NUM*	*O-N-n.LOCATION*	*O-P-p.Goal*	*B-N-n.LOCATION*	*I_*	*O-PUNCT*
O-V-v.motion	*O-P-p.Direction*	*O-NUM*	*O-N-n.RELATION*	*O-P-p.Goal*	*B-N-n.GROUP*	*I_*	*O-PUNCT*

beware	they	will	rip	u	off
O-V-v.cognition	*O-PRON*	*O-AUX*	**O-V-v.contact**	*o-PRON*	*I_*
O-V-v.cognition	*O-PRON*	*O-AUX*	*B-V.VPC.full-v.social*	*o-PRON*	*I_*

Figure 3: Selected examples where the model without MWE constraints (first row under each sentence) produces a structurally invalid tagging. Incorrect tags are red; the ones that render the tagging structurally invalid are bold. The last row under each sentence is the gold annotation, and the middle row (if different from gold) is the model prediction with MWE constraints. (The first sentence ends with a period, omitted for brevity.)

often better than using predicted POS/lemmas. Removing the MWE constraints yields models with slightly higher overall tag accuracy, but results in invalid segmentations for a large proportion of sentences: 14% of STREUSLE sentences in the fully unconstrained model and 17% of sentences if only predicted POS and lemmas are used for constraints.

Three sentences out of those 17% appear in figure 3. The first shows both an omission of a “B-” tag needed to start an MWE (“new”) and a false positive gap without members of an MWE on either side (“me”). When the full set of constraints is used, the gold tagging is recovered. In the second sentence, there is a false positive yet structurally valid MWE (“Go down”) as well as an invalid start to an MWE that is never continued (“Super”), perhaps because it is rare for a number to continue an MWE (this happens <20 times in the entire corpus). Finally, in the third sentence, the model constrained only by POS and lemma is inclined toward the literal meaning of “rip”, whereas the MWE-constrained model recovers the gappy verb-particle construction “rip off”. Naturally, in other sentences, the MWE-constrained model sometimes suffers from false positive or false negative MWEs, but always produces a coherent segmentation.

4 Related Work

The computational study of MWEs has a long history (Sag et al., 2002; Diab and Bhutada, 2009; Baldwin and Kim, 2010; Ramisch, 2015; Qu et al., 2015; Constant et al., 2017; Bingel and Søgaard, 2017; Shwartz and Dagan, 2019), as does supersense tagging (Segond et al., 1997; Ciaramita and Altun, 2006). Vincze et al. (2011) developed a sequence tagger for both MWEs and named entities in English. Schneider and Smith (2015); Schneider et al. (2016) featured joint tagging of MWEs and noun and verb supersenses with feature-based sequence models. Richardson (2017) trained such a model on STREUSLE 3.0 as a noun, verb, and preposition supersense tagger (without modeling MWEs). For preposition supersenses, Gonen and Goldberg (2016) incorporated multilingual cues; Schneider et al. (2018) experimented with feature-based and neural classifiers; and Liu et al. (2019), modeling supersense disambiguation of single-word prepositions only, found pretretrained contextual embeddings to be much more effective even with simple linear probing models.

5 Conclusion

We study the lexical semantic recognition task defined by the STREUSLE corpus, which involves joint MWE identification and coarse-grained (supersense) disambiguation of noun, verb, and preposition expressions; this task subsumes and unifies the previous PARSEME and DiMSUM evaluations. We develop a strong baseline neural sequence model, and see encouraging results on the task. Furthermore, zero-shot out-of-domain evaluation of our baselines on partial versions of the task yields scores comparable to the fully-supervised in-domain state of the art.

Acknowledgments

We are grateful to anonymous reviewers as well as members of the NERT lab for their feedback on this work. This research was supported in part by NSF award IIS-1812778 and grant 2016375 from the United States–Israel Binational Science Foundation (BSF), Jerusalem, Israel. NL is supported by an NSF Graduate Research Fellowship under grant number DGE-1656518.

References

Mostafa Abdou, Artur Kulmizev, Vinit Ravishankar, Lasha Abzianidze, and Johan Bos. 2018. What can we learn from semantic tagging? In *Proc. of EMNLP*, pages 4881–4889, Brussels, Belgium.

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *Proc. of IWCS*, Montpellier, France.

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*, pages 267–292. CRC Press, Taylor and Francis Group, Boca Raton, FL.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. Technical Report LDC2012T13, Linguistic Data Consortium, Philadelphia, PA.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proc. of EACL*, pages 164–169, Valencia, Spain.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proc. of COLING*, pages 3531–3541, Osaka, Japan.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proc. of EMNLP*, pages 594–602, Sydney, Australia.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: a survey. *Computational Linguistics*, 43(4):837–892.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pages 4171–4186, Minneapolis, Minnesota.

Mona Diab and Pravin Bhutada. 2009. Verb noun construction MWE token classification. In *Proc. of MWE*, pages 17–22, Suntec, Singapore.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or surface-syntactic universal dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74, Brussels, Belgium. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2016. Semi supervised preposition-sense disambiguation using multilingual data. In *Proc. of COLING*, pages 2718–2729, Osaka, Japan.

Oliver Hellwig. 2017. Coarse semantic classification of rare nouns using cross-lingual data and recurrent neural networks. In *Proc. of IWCS*, Montpellier, France.

Jena D. Hwang, Archna Bhatia, Na-Rae Han, Tim O'Gorman, Vivek Srikumar, and Nathan Schneider. 2017. Double trouble: the problem of construal in semantic annotation of adpositions. In *Proc. of *SEM*, pages 178–188, Vancouver, Canada.

Angelika Kirilin, Felix Krauss, and Yannick Versley. 2016. ICL-HD at SemEval-2016 Task 10: Improving the Detection of Minimal Semantic Units and their Meanings with an ontology and word embeddings. In *Proc. of SemEval*, pages 937–945, San Diego, California.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289, Williamstown, MA, USA.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proc. of NAACL-HLT*, pages 1073–1094, Minneapolis, Minnesota.

Héctor Martínez Alonso, Anders Johannsen, Sussi Olsen, Sanni Nimb, Nicolai Hartvig Sørensen, Anna Braasch, Anders Søgaard, and Bolette Sandford Pedersen. 2015. Supersense tagging for Danish. In *Proc. of NODALIDA*, pages 21–29, Vilnius, Lithuania.

Luka Nerima, Vasiliki Foufi, and Éric Wehrli. 2017. Parsing and MWE detection: Fips at the PARSEME shared task. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 54–59, Valencia, Spain. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proc. of LREC*, pages 4027–4036, Marseille, France.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proc. of EMNLP*, pages 1532–1543, Doha, Qatar.

Davide Picca, Alfio Massimiliano Gliozzo, and Massimiliano Ciaramita. 2008. Supersense Tagger for Italian. In *Proc. of LREC*, pages 2386–2390, Marrakech, Morocco.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency parsing from scratch. In *Proc. of CoNLL*, pages 160–170, Brussels, Belgium.

Likun Qiu, Yunfang Wu, Yanqiu Shao, and Alexander Gelbukh. 2011. Combining contextual and structural information for supersense tagging of Chinese unknown words. In *Computational Linguistics and Intelligent Text Processing: Proceedings of the 12th International Conference (CICLing'11)*, volume 6608 of *Lecture Notes in Computer Science*, pages 15–28. Springer, Berlin.

Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, Nathan Schneider, and Timothy Baldwin. 2015. Big data small data, in domain out-of domain, known word unknown word: the impact of word representations on sequence labelling tasks. In *Proc. of CoNLL*, pages 83–93, Beijing, China.

Carlos Ramisch. 2015. Multiword expressions acquisition. *A Generic and Open Framework. Cham: Springer International Publishing*.

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME Shared Task on Automatic Identification of Verbal Multiword Expressions. In *Proc. of LAW-MWE-CxG-2018*, pages 222–240, Santa Fe, New Mexico, USA.

Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk, Marie Candito, Ashwini Vaidya, Verginica Barbu Mititelu, Archna Bhatia, Uxoa Iñurrieta, Voula Giouli, Tunga Güngör, Menghan Jiang, Timm Lichte, Chaya Liebeskind, Johanna Monti, Renata Ramisch, Sara Stymne, Abigail Walsh, and Hongzhi Xu. 2020. Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118, online. Association for Computational Linguistics.

Oliver Ethan Richardson. 2017. *Joint prediction of supersense relations*. Bachelor's thesis, University of Utah, Salt Lake City, Utah.

Omid Rohanian, Shiva Taslimipoor, Samaneh Kouchaki, Le An Ha, and Ruslan Mitkov. 2019. Bridging the gap: Attending to discontinuity in identification of multiword expressions. In *Proc. of NAACL-HLT*, pages 2692–2698, Minneapolis, Minnesota.

Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: a pain in the neck for NLP. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 189–206. Springer, Berlin.

Agata Savary, Carlos Ramisch, Silvio Ricardo Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The PARSEME Shared Task on Automatic Identification of Verbal Multiword Expressions. In *Proc. of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. 2014a. Discriminative lexical semantic segmentation with gaps: running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

Nathan Schneider, Dirk Hovy, Anders Johannsen, and Marine Carpuat. 2016. SemEval-2016 Task 10: Detecting Minimal Semantic Units and their Meanings (DiMSUM). In *Proc. of SemEval*, pages 546–559, San Diego, California, USA.

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. Comprehensive supersense disambiguation of English prepositions and possessives. In *Proc. of ACL*, pages 185–196, Melbourne, Australia.

Nathan Schneider, Behrang Mohit, Chris Dyer, Kemal Oflazer, and Noah A. Smith. 2013. Supersense tagging for Arabic: the MT-in-the-middle attack. In *Proc. of NAACL-HLT*, pages 661–667, Atlanta, Georgia, USA.

Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta Conrad, and Noah A. Smith. 2014b. Comprehensive annotation of multiword expressions in a social web corpus. In *Proc. of LREC*, pages 455–461, Reykjavík, Iceland.

Nathan Schneider and Noah A. Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In *Proc. of NAACL-HLT*, pages 1537–1547, Denver, Colorado.

Frédérique Segond, Anne Schiller, Gregory Grefenstette, and Jean-Pierre Chanod. 1997. An experiment in semantic tagging using hidden Markov model tagging. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications: ACL/EACL-97 Workshop Proceedings*, pages 78–81, Madrid, Spain.

Yutaro Shigeto, Ai Azuma, Sorami Hisamoto, Shuhei Kondo, Tomoya Kouse, Keisuke Sakaguchi, Akifumi Yoshimoto, Frances Yung, and Yuji Matsumoto. 2013. Construction of English MWE dictionary and its application to POS tagging. In *Proc. of the 9th Workshop on Multiword Expressions*, pages 139–144, Atlanta, Georgia, USA.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Shiva Taslimipoor, Omid Rohanian, and Le An Ha. 2019. Cross-lingual transfer learning and multitask learning for capturing multiword expressions. In *Proc. of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 155–161, Florence, Italy.

Veronika Vincze, István Nagy T., and Gábor Berend. 2011. Multiword expressions and named entities in the Wiki50 corpus. In *Proc. of RANLP*, pages 289–295, Hissar, Bulgaria.

Abigail Walsh, Claire Bonial, Kristina Geeraert, John P. McCrae, Nathan Schneider, and Clarissa Somers. 2018. Constructing an annotated corpus of verbal MWEs for English. In *Proc. of LAW-MWE-CxG-2018*, pages 193–200, Santa Fe, New Mexico, USA.

Yilun Zhu, Yang Liu, Siyao Peng, Austin Blodgett, Yushi Zhao, and Nathan Schneider. 2019. Adpositional Supersenses for Mandarin Chinese. In *Proc. of SCiL*, volume 2, pages 334–337, New York, NY, USA.

Appendices

A Lexical categories in STREUSLE

Lexcat	SS	Definition	Lexcat	Definition
`N`	n.*	noun, common or proper	`NUM`	number
`PRON.POSS`	p.*	possessive pronoun	`PRON`	non-possessive pronoun
`POSS`	p.*	possessive clitic (*'s*)	`ADJ`	adjective
`P`	p.*	adposition	`ADV`	adverb
`PP`	p.*	(idiomatic) adpositional phrase MWE	`DET`	determiner
`INF.P`	p.*	semantically annotatable infinitive marker	`INF`	nonsemantic infinitive marker
`V`	v.*	**single-word** full verb or copula	`AUX`	auxiliary, not copula
`V.VID`	v.*	**MWE:** verbal idiom	`DISC`	discourse/pragmatic expression
`V.VPC.full`	v.*	**MWE:** full verb-particle construction	`CCONJ`	coordinating conjunction
`V.VPC.semi`	v.*	**MWE:** semi verb-particle construction	`SCONJ`	subordinating conjunction
`V.LVC.full`	v.*	**MWE:** full light verb construction	`INTJ`	interjection
`V.LVC.cause`	v.*	**MWE:** causative light verb construction	`SYM`	symbol
`V.IAV`	v.*	**MWE:** idiomatic adpositional verb	`PUNCT`	punctuation
			`X`	foreign or nonlinguistic

Table 3: Lexcats (lexical categories) that are annotated for strong lexical units, i.e., single-word expressions or strong MWEs. Weak MWEs are treated as compositional and thus do not receive a holistic lextag or supersense. ***Left:*** Lexcats that require supersenses of the class designated in the second column: nominal (n.*), verbal (v.*), or adpositional/possessive (p.*). Verbal MWEs are syntactically subtyped in the lexcat, and the simple `V` lexcat applies to non-MWEs only. ***Right:*** Lexcats that are incompatible with supersenses. Most of these are defined in line with Universal POS tag definitions, but may also apply to MWEs. Definitions come from `https://github.com/nert-nlp/streusle/blob/master/CONLLULEX.md`.

B Baseline Implementation Details

Table 4 lists the hyperparameter values we found by tuning on the STREUSLE development set, with BERT pre-trained contextualized embeddings (large-cased; Devlin et al., 2019), predicted POS tags and lemmas. BERT parameters are not fine-tuned.

BiLSTM #layers	2
BiLSTM total dim. per layer	512
Learning rate	0.001
Batch size	64

Table 4: Hyperparameter values.

Our tagger uses the BERT (large, cased) pretrained model to produce input word representations; these input word representations are a learned scalar mixture of the BERT representations, following observations that the topmost layer of BERT is highly attuned to the pretraining task and generalizes poorly (Liu et al., 2019). The representation for a token is taken to be BERT output corresponding to its first wordpiece representation. We freeze the BERT representations during training.

The word representations from the frozen BERT contextualizer are then fed into a 2-layer bidirectional LSTM with 256 hidden units in each direction. The LSTM outputs then are projected into the label space with a learned linear function, and a linear chain conditional random field produces the final output.

For training, we minimize the negative log-likelihood of the tag sequence with the Adam optimizer, using a batch size of 64 sequences and a learning rate of 0.001.

We train our model for 75 epochs, and gradient norms are rescaled to a maximum of 5.0. We apply early stopping with a patience of 25 epochs. Our model is implemented in the AllenNLP framework (Gardner et al., 2018).

In our ablated models that use GloVe vectors and character-level CNNs instead of BERT, we use 200 output filters with a window size of 5 in the CNN. The input to the CNN are 64-dimensional character embeddings.

C Per-Lexcat STREUSLE Results

Table 5 shows STREUSLE test set results for the BERT tagger with only MWE constraints (no POS/lemma constraints), broken down by lexical category. The numbers reported here differ from the evaluation in table 1—these metrics are calculated by extracting the predicted and gold spans, and then computing an exact-match F1 measure between the predicted and gold sets.

Frequency counts are for STREUSLE-test; OOV token rates are relative to STREUSLE-train. Examples are lemmatized lexical units ("lexlemmas"). Lexlemmas are used to calculate OOV rates.

Lexcat	Example	# Gold	% OOV	P	R	F	Lexcat	Example	# Gold	% OOV	P	R	F
`N`	*food*	946	24.7%	85.5	88.9	87.2	`NUM`	*five*	41	17.1%	92.9	95.1	94.0
`PRON.POSS`	*my*	94	0.0%	98.9	93.6	96.2	`PRON`	*it*	393	0.0%	95.1	98.2	96.6
`POSS`	*'s*	1	0.0%	100.0	100.0	100.0	`ADJ`	*best*	532	8.9%	85.8	94.0	89.7
`P`	*with*	322	0.3%	88.0	93.2	90.5	`ADV`	*extremely*	358	2.1%	91.7	92.2	91.9
`PP`	*by far*	18	0.0%	87.5	77.8	82.4	`DET`	*the*	376	0.0%	92.4	96.8	94.5
`INF.P`	*to* see	20	0.0%	87.0	100.0	93.0	`INF`	*to*	36	0.0%	91.9	94.4	93.2
`V`	*go*	587	3.5%	90.0	95.4	92.6	`AUX`	*have*	160	0.0%	95.7	96.2	96.0
`V.VID`	*take time*	24	0.0%	64.3	37.5	47.4	`DISC`	*thanks*	21	4.5%	63.6	66.7	65.1
`V.VPC.full`	*turn out*	11	0.0%	58.3	63.6	60.9	`CCONJ`	*and*	204	0.0%	95.3	99.5	97.4
`V.VPC.semi`	*add on*	5	0.0%	50.0	60.0	54.5	`SCONJ`	*lest*	21	4.8%	90.5	90.5	90.5
`V.LVC.full`	*have fun*	8	0.0%	60.0	37.5	46.2	`INTJ`	*hey*	17	35.3%	78.6	64.7	71.0
`V.LVC.cause`	*give chance*	1	0.0%	0.0	0.0	0.0	`SYM`	*:)*	12	0.0%	100.0	75.0	85.7
`V.IAV`	*treat to*	17	5.9%	81.8	52.9	64.3	`PUNCT`	.	597	0.3%	99.0	99.7	99.3
							`X`	*etc*	2	50.0%	0.0	0.0	0.0

Table 5: STREUSLE test set results for the BERT-based tagger with only MWE constraints (no POS/lemma constraints), broken down by lexcat. The numbers reported here differ from the evaluation in table 1—these metrics are calculated by extracting the predicted and gold spans, and then computing an exact-match F1 measure between the predicted and gold sets.

D Per-VMWE Category PARSEME Results

Table 6 shows PARSEME (English) test set results for the BERT tagger with only MWE constraints (no POS/lemma constraints), broken down by VMWE category.

Frequency counts are for PARSEME-EN-test and reflect the number of gold MWEs; OOV token rates are relative to STREUSLE-train. Examples are lemmatized lexical units ("lexlemmas"). Lexlemmas are used to calculate OOV rates.

PARSEME 1.1				MWE-based			Token-based		
VMWEs (EN-test)	Example	# Gold	% OOV	P	R	F	P	R	F
`V.VID`	*tide turn*	79	80.6%	8.8	17.7	11.8	11.8	20.9	15.1
`V.VPC.full`	*bring in*	146	44.3%	41.8	59.6	49.2	43.3	63.7	51.5
`V.VPC.semi`	*speak up*	26	61.5%	12.7	30.8	18.0	12.5	30.8	17.8
`V.LVC.full`	*make promise*	166	90.5%	30.8	4.8	8.3	38.2	6.1	10.6
`V.LVC.cause`	*yield result*	36	100.0%	0.0	0.0	0.0	0.0	0.0	0.0
`V.IAV`	*turn into*	44	52.2%	30.4	38.6	34.0	29.1	37.8	32.9
`V.MVC`	*cross examine*	4	80.0%	0.0	0.0	0.0	0.0	0.0	0.0

Table 6: PARSEME (English) test set results for the BERT tagger with only MWE constraints (no POS/lemma constraints), broken down by VMWE category.

E VMWE Performance in STREUSLE vs. PARSEME

PARSEME appears to be a much more challenging task, even considering just the VMWE identification performance in STREUSLE: in the main text, compare BERT model F-scores of 64% in STREUSLE versus 40% in PARSEME (where the state-of-the-art result is 42%). Why is this the case? We suspect at least three factors are at play:

- Substantial domain shift: PARSEME covers a wide range of genres, including literary genres, which is likely to contribute to lower precision and recall in general.
- Base rate of MWEs: STREUSLE contains about 10 times as many MWEs per word as PARSEME, in part due to the comprehensive nature of MWE annotation in STREUSLE. Considering just verbal MWEs per word, STREUSLE-train has 763/44,815 = 1.7% and STREUSLE-test has 66/5,381 = 1.2%, whereas PARSEME-test has 501/71,002 = 0.7%. So it is not surprising that the STREUSLE-trained tagger would overpredict VMWEs in PARSEME. Note that precision is lower than recall overall and for most VMWE subtypes.
- OOV rate: MWE identification of lexical items unseen in training is generally more challenging. We see above that the VMWE vocabularies of STREUSLE and PARSEME are largely disjoint, with OOV rates above 50% for most subtypes. This would be expected to mainly impact recall, and in fact, the higher the OOV rate, in general the lower the recall. In particular, recall is much lower than precision for the LVC.full subtype, with an OOV rate of 90.5%, suggesting that it is able to correctly identify some known LVCs but unable to generalize to new ones. The 8 instances correctly identified had, in fact, been seen in training.