UFRGS&LIF at SemEval-2016 Task 10: Rule-Based MWE Identification and Predominant-Supersense Tagging

Silvio Ricardo Cordeiro[1,2] and **Carlos Ramisch**[2] and **Aline Villavicencio**[1]
[1] Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil
[2]Aix Marseille Université, CNRS, LIF UMR 7279
{srcordeiro,avillavicencio}@inf.ufrgs.br
carlos.ramisch@lif.univ-mrs.fr

Abstract

This paper presents our approach towards the SemEval-2016 Task 10 – Detecting Minimal Semantic Units and their Meanings. Systems are expected to provide a representation of lexical semantics by (1) segmenting tokens into words and multiword units and (2) providing a supersense tag for segments that function as nouns or verbs. Our pipeline rule-based system uses no external resources and was implemented using the mwetoolkit. First, we extract and filter known MWEs from the training corpus. Second, we group input tokens of the test corpus based on this lexicon, with special treatment for non-contiguous expressions. Third, we use an MWE-aware predominant-sense heuristic for supersense tagging. We obtain an F-score of 51.48% for MWE identification and 49.98% for supersense tagging.

1 Introduction

Accurate segmentation and semantic disambiguation of minimal text units is a major challenge in the general pipeline of NLP applications. A machine translation system, for example, needs to decide what is the intended meaning for a given word or phrase in its context, so that it may translate it into an equivalent meaning in the target language.

While determining the meaning of single words is a difficult task on its own, the problem is compounded by the pervasiveness of Multiword Expressions (MWEs). MWEs are semantic units that span over multiple lexemes in the text (e.g. dry run, look up, fall flat). Their meaning cannot be inferred by applying regular composition rules on the meanings of their component words. The task of semantic tagging is thus deeply intertwined with the identification of multiword expressions.

This paper presents our solution to the DiMSUM shared task (Schneider et al., 2016), where the evaluated systems are expected to perform both semantic tagging and multiword identification. Our pipeline system first detects and groups MWEs and then assigns supersense tags, as two consecutive steps. For MWE identification, we use a task-specific instantiation of the mwetoolkit (Ramisch, 2015), handling both contiguous and non-contiguous MWEs with some degree of customization (Cordeiro et al., 2015). Additionally, MWE type-level candidates are extracted without losing track of their token-level occurrences, to guarantee that all the MWE occurrences learned from the training data are projected onto the test corpus. For semantic tagging we adopted a predominant-sense heuristic.

In the remainder of this paper, we present related work (§ 2), then we present and discuss the results of the MWE identification subsystem (§ 3) and of the supersense tagging subsystem (§ 4). We then conclude and share ideas for future improvements (§ 5).

2 Related Work

Practical solutions for rule-based MWE identification include tools like jMWE (Kulkarni and Finlayson, 2011), a library for direct lexicon projection based on preexisting MWE lists. Finite-state transducers can also be used to take into account the internal morphology of component words and perform efficient tokenization based on MWE dictionaries (Savary, 2009). The problem of MWE identification

Proceedings of SemEval-2016, pages 910–917,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

has also been modeled using supervised machine learning. Probabilistic MWE taggers usually encode the data using a begin-inside-outside scheme and learn CRF-like taggers on it (Constant and Sigogne, 2011; Schneider et al., 2014). The mwetoolkit (Ramisch, 2015) provides command-line programs that allow one to discover new MWE candidate lists, filter them and project them back on text according to some parameters. Our system uses the latter as basis for MWE identification.

Word sense disambiguation (WSD) methods can be roughly classified into knowledge-based, supervised and unsupervised. Knowledge-based methods use lexico-semantic taxonomies like WordNet to calculate the similarity between context and target words (Lesk, 1986). Supervised approaches generally use context-sensitive classifiers (Cabezas et al., 2001). Unsupervised approaches using clustering and distributional similarity (Brody and Lapata, 2008; Goyal and Hovy, 2014) can also be employed for WSD. Both supervised and unsupervised WSD techniques have also been used to distinguish literal from idiomatic uses of MWEs (Fazly et al., 2009; Diab and Bhutada, 2009). Nonetheless, systematically choosing the most frequent sense is a surprisingly good baseline, not always easy to beat (McCarthy et al., 2007; Navigli, 2009). This was also verified for MWE disambiguation (Uchiyama et al., 2005). Thus, in this work, we implemented a simple supervised predominant-sense heuristic and will investigate more sophisticated WSD techniques as future work.

3 MWE Identification

Our MWE identification algorithm uses 6 different rule configurations, targeting different MWE classes. Three of these are based on data from the training corpus, while the other three are unsupervised. The parameters of each configuration are optimized on a held-out development set, consisting of ⅑ of the training corpus. The final system is the union of all configurations.[1]

For the 3 supervised configurations, annotated MWEs are extracted from the training data and then filtered: we only keep combinations that have been annotated often enough in the training corpus. In other words, we keep MWE candidates whose proportion of annotated instances with respect to all occurrences in the training corpus is above a threshold t, discarding the rest. The thresholds were manually chosen based on what seemed to yield better results on the development set. Finally, we project the resulting list of MWE candidates on the test data, that is, we segment as MWEs the test token sequences that are contained in the lexicon extracted from the training data. These configurations are:

CONTIG Contiguous MWEs annotated in the training corpus are extracted and filtered with a threshold of $t = 40\%$. That is, we create a lexicon containing all contiguous lemma+POS sequences for which at least 40% of the occurrences in the training corpus were annotated. The resulting lexicon is projected on the test corpus whenever that contiguous sequence of words is seen.

GAPPY Non-contiguous MWEs are extracted from the training corpus and filtered with a threshold of $t = 70\%$. The resulting MWEs are projected on the test corpus using the following rule: an MWE is deemed to occur if its component words appear sequentially with at most a total of 3 gap words in between them.

NOUN^2-KN Collect all noun-noun sequences in the test corpus that also appear at least once in the training corpus (known compounds), and filter them with a threshold of $t = 70\%$. The resulting list is projected onto the test corpus.

We further developed 3 additional configurations based on empirical findings. We identify MWEs in the test corpus based on POS-tag patterns, without any filtering (and thus without looking at the training corpus)[2]:

NOUN^2-UKN Collect all noun-noun sequences in the test corpus that never appear in the training corpus (unknown compounds), and project all of them back on the test corpus.

PROPN$^{2..\infty}$ Collect sequences of two or more contiguous words with POS-tag PROPN and project all of them back onto the test corpus.

[1] When there is an overlap, we favor longer MWEs.

[2] For NOUN^2-UKN, we exclude known compounds, as otherwise that would undo the filtering work done by NOUN^2-KN.

VP Collect verb-particle candidates and project them back onto the test corpus. A verb-particle candidate is a pair of words under these constraints: the first word must have POS-tag `VERB` and cannot have lemma *go* or *be*. The two words may be separated by a `N`[3] or `PROPN`. The second word must be in a list of frequent non-literal particles[4]. Finally, the particle must be followed by a word with one of these POS-tags: `ADV`, `ADP`, `PART`, `CONJ`, `PUNCT`. Even though we might miss some cases, this final delimiter avoids capturing regular verb-PP sequences.

Table 1 presents the results for each isolated configuration (evaluated on the test corpus, with all MWEs). These results are calculated based on the fuzzy metrics of the shared task (Schneider et al., 2014), where partial MWE matches are taken into account. Our final MWE identification system is the union of all rule configurations described above. The final recall of the system is not the sum of coverage values because MWE candidate lexicons may overlap (multiple configurations may have identified the same MWE).

Configuration	Precision	Coverage
CONTIG	57.9%	11.6%
GAPPY	36.0%	0.9%
NOUN^2-KN	100.0%	1.6%
NOUN^2-UKN	80.2%	18.9%
PROPN$^{2..\infty}$	96.0%	8.5%
VP	71.2%	4.2%

Table 1: Precision and coverage per MWE annotation. Coverage is the recall of each configuration applied independently.

3.1 Error Analysis

Table 2 presents the system results for the most common POS-tag sequences in the test corpus, using an exact match (a MWE is either correct or incorrect). Overall results are presented in both exact and fuzzy metrics.

`N_N` errors Since our system looks for all occurrences of adjacent noun-noun pairs, we obtain a high recall for `N_N` compounds. The most common false positive errors are presented below.

POS-tags	Precision	Recall	F$_1$
`N_N`	170/278 = 61%	170/181 = 94%	74.0%
`VERB_ADP`	43/60 = 72%	43/73 = 59%	64.9%
`ADJ_N`	5/6 = 83%	5/69 = 7%	12.9%
`PROPN_PROPN`	65/82 = 79%	65/66 = 98%	87.5%
`VERB_PART`	31/37 = 84%	31/49 = 63%	72.0%
`PROPN_N`	1/1 = 100%	1/34 = 3%	5.8%
`N_N_N`	0/0 = 100%	0/22 = 0%	0.0%
`ADP_N`	10/14 = 71%	10/22 = 45%	55.1%
`VERB_N`	1/5 = 20%	1/16 = 6%	9.2%
`DET_N`	4/23 = 17%	4/16 = 25%	20.2%
`ADJ_N_N`	0/0 = 100%	0/11 = 0%	0.0%
Overall (exact)	364/613 = 59%	364/837 = 43%	50.2%
Overall (fuzzy)	460/635 = 72%	461/1115 = 41%	52.6%

Table 2: MWE identification results on test set per POS-tag.

- **Not in the same phrase** In 19 cases, our system has identified two `N`s that are not in the same phrase; e.g. **when I have a problem customer services don't want to know*. In order to realize that these nouns are not related, we would need parsing information. Nonetheless, it is not clear whether an off-the-shelf parser could solve these ambiguities in the absence of punctuation.

- **Partial `N_N_N`** 17 cases have been missed due to only the first two nouns in the MWE being identified; e.g. **Try the memory foam pillows!* – instead of *memory foam pillows*.

- **Partial `ADJ_N_N`** 10 cases have been missed; e.g. **My sweet pea plants arrived 00th May completely dried up and dead!* – instead of *sweet pea plants*. These cases are a consequence of the fact that we do not look for adjective-noun pairs (see `ADJ_N` errors below).

- **Compositional `N_N`** In 24 cases, our system identified a compositional compound; e.g. **Quality gear guys, excellent!* Semantic features would be required to filter such cases out.

- **Questionable `N` tags** 10 false noun compounds were found due to words such as *today* being tagged as nouns (e.g. **I'm saving gas today*). Another 5 cases had adjectives classified as nouns: **Maybe this is a kind of an artificial way to read an e-book.*

[3] In the remainder of the paper, we abbreviate the POS tag `NOUN` as `N`.

[4] The 13 most frequent non-literal particles: *about*, *around*, *away*, *back*, *down*, *in*, *into*, *off*, *on*, *out*, *over*, *through*, *up* (Sinclair, 2012).

VERB_ADP errors Most of the VERB_ADP expressions were caught by the VP configuration, but we still had some false negatives. In 7 cases, the underlying particle was not in our list (e.g. *I regret ever going near their store*), while in 9 other cases, the particle was followed by a noun phrase (e.g. *Givin out Back shots*). 5 of the missed MWEs could have been found by accepting the particle to be followed by a SCONJ, or to be followed by the end of the line as delimiters. Most of the false positives were due to the verb being followed by an indirect object or prepositional phrase. We believe that disambiguating these cases would require valency information, either from a lexicon or automatically acquired from large corpora (Preiss et al., 2007).

ADJ_N errors While the few ADJ_N pairs that our system identified were usually correct MWEs, most of the annotated cases were missed. This is because we do not specifically look for adjective-noun pairs, due to the high likelihood of them being compositional. For example, a simple ADJ_N annotation scheme (as performed in NOUN^2-UKN) would have achieved a precision of only $69/505 = 14\%$.

Out of all annotated sentences, in 23 cases the noun is transparent, and we could replace the adjective by a synonym; e.g. *I guess people are going again next week, do you think you'll go?* (which could be replaced by *the following week*). In another 17 cases, the noun is transparent and the adjective suggestive of the global meaning, even though it is fixed; e.g. *23 is the lucky number* (but not *_fortunate number_, albeit related to *luck*).

These cases could be dealt with using fixedness tests such as substitution and permutation (Fazly et al., 2009; Ramisch et al., 2008).

PROPN_PROPN errors Since our system looks for all occurrences of adjacent PROPN pairs, we obtain near-perfect recall for PROPN_PROPN compounds. Most false positives were caused by possessives or personal titles, which were annotated as part of the MWE in the gold standard.

VERB_PART errors The results for VERB_PART are similar to the ones found for VERB_ADP: 3 false negatives are due to the particle not being in our list, and in another 7 cases they are followed by a noun phrase. Additionally, in 6 cases the particle was followed by a verb (e.g. *Stupid Kilkenny didn't get to meet @Royseven*). 4 false positives were CONTIG cases of go to being identified as a MWE (e.g. *_In my mother's day, she didn't go to college_). In the training corpus, this MWE had been annotated 57% of the time, but in future constructions (e.g. *Definitely not going to purchase a car from here*). Canonical forms would be easy to model with a specific contextual rule of the form *going to* verb.

PROPN_N errors While the few PROPN_N pairs we found were all correct MWEs, most of the annotated cases were missed. These cases did not earn special attention during the development of the system due to an incorrectly perceived infrequency. However, using only an annotation scheme such as NOUN^2-UKN, we could have achieved a precision of 72% for these MWEs.

N_N_N errors The occurrence of N_N_N sequences is rare in the training corpus, and we did not specifically look for them, which explains our recall of 0%. By annotating the longest sequence of Ns in the corpus (NOUN$^{2..\infty}$), we could have obtained a precision of 56% and recall of 91% for N_N_N. The precision of N_N would also increase to 70% (with a recall of 93%). If we then replace NOUN^2 by NOUN$^{2..\infty}$, the full-system's F-score increases to 56.23%.

ADP_N errors The false positives were ambiguous determinerless PPs that can be compositional or not according to the context. For instance, the system identified *_Try them all, in order_ after seeing *The Big Lebowski is in order tonight*. False negatives were mainly due to threshold-based filters, like *at all* and *in peace*. Unsupervised MWE discovery on large corpora using context-sensitive association measures could have helped in these cases.

VERB_N errors We only generated 4 false positives, which look like light-verb constructions missed by the annotators (*give ride*, *place order*) False negatives include 8 cases of gerunds POS-tagged as verbs (e.g. *to listen to flying saucers*), which are actualy similar to ADJ_N cases discussed above. We also found 7 false negatives, mainly light-verb constructions, that were not present in the training corpus (*take place, take control*).

`DET_N` errors 8 false negatives were compositional time adjuncts (e.g. *this morning, this season*). False positives are mainly cases that seem inconsistent between training and test data concerning frequent quantifiers (e.g. *a lot, a bit, a couple*).

Noun compounds (two or more `N`s in a row) account for a significant proportion of MWEs in the training corpus ($^{601}/_{4232} = 14\%$) and an even larger amount of the testing corpus ($^{203}/_{837} = 24\%$). The NOUN^2 rule sets were essential to obtaining good results. If we remove NOUN^2 from our system, its global performance would drop to a fuzzy $\mathrm{F}_1 = 33.79\%$.

The domain of the corpus does not seem to have a great influence on our method's performance. Our lowest performance is on the Reviews subcorpus (fuzzy $\mathrm{F}_1 = 49.57\%$) and our best performance is on TED (fuzzy $\mathrm{F}_1 = 56.76\%$).

Some of the missed MWEs are questionable and we feel that our system should not annotate them. These include regular verbal chains (*shouldn't have, have been*), infinitival and selected preposition *to* (*to take, go to*) and compositional noun phrases (*this Saturday*). Fortunately, these cases correspond to a small proportion of the data.

4 Supersense Tagging

Supersense tagging takes place after MWE identification. Sense tags are coarse top-level Wordnet synsets. The tagset for nouns and verbs has respectively 26 and 15 supersense tags. We use a predominant-sense heuristic to perform WSD.

Before tagging the test data, our system collects all annotated supersense tags from MWEs in the training corpus. We create a mapping with entries of the form $(\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_N) \mapsto \mathrm{S}$, where each MWE component $\mathrm{w}_i = (\mathrm{lemma}_i, \mathrm{POStag}_i)$. This mapping indicates the most frequent tag S associated a given MWE. Single words are treated as length-1 MWEs and are also added to this mapping.

The supersense tagging algorithm then goes through all segmented units (MWEs or single words) in the test corpus and annotates them according to the most common tag seen in the training set. If a tag has not been seen for a given word or MWE, we do not tag it at all. This heuristic is very simple and not very realistic. Nonetheless, it allowed us to have a minimal supersense tagger quickly and then focus on accurate MWE identification as the main contribution of our system.

4.1 Error Analysis

Tables 3 and 4 show the confusion matrices of our system for the 10 most common tags. Each row corresponds to a gold tag and contains the distribution of predicted tags. The perfect system would have numbers only in the main diagonal and zeros everywhere else. The skewed distribution of supersense tags makes our simple heuristic quite effective when the MWE/word has been observed in the training data.

Known nouns seem easy to tag. Most of our errors come from the fact that we did not observe instances of a noun in the training data, and thus did not assign it any tag (column "skipped"). Some distinctions seem harder than others due to similar semantic classes: attributive/cognition and event/time.

The occurrence of verbs in the training data is less of a problem than their polysemy. Stative verbs correspond to the large majority of verbs in the dataset. This is magnified by the nature of the corpus: reviews tend to use stative verbs to talk about product characteristics, tweets often use them to describe the state of the author. While very frequent, stative verbs are also difficult to disambiguate: most false negatives were tagged as change verbs while most false positives were tagged as social verbs. Some distinctions seem extremely hard to make, specially for less frequent supersense tags like contact/motion and perception/cognition.

5 Conclusions and Future Work

We developed a simple rule-based system that was able to obtain competitive results. Its main advantage is that it was very quick to implement in the context of the generic framework of the `mwetoolkit`. The system is freely available as part of the official `mwetoolkit` release.[5] The main limitation of our system is that it cannot properly take unseen MWEs into account and generalize from seen instances. Moreover, most of our rule sets are highly language dependent.

Ideas for future improvements include:

[5] `http://mwetoolkit.sourceforge.net`

Gold tag	PERS	ARTI	COMM	ACT	GROU	TIME	COGN	ATTR	LOCA	EVEN	skipped	total
n.person	157	1			8			3			238	413
n.artifact	2	103	6	3	6		5	3	8		187	337
n.communic	2	4	128	4	2	1	6	2	2	1	119	284
n.act	1	9	5	111	4	1	9			14	83	256
n.group	6	4	2	1	54				2	1	130	205
n.time				1		114	2			6	56	180
n.cognition	1	4	3	3			48	1		5	51	130
n.attribute	3		1	3	1		30	10			53	130
n.location		5		2	9	7	2		23	1	45	99
n.event	1					16	1			28	31	84
not-a-noun	123	18	23	12	44	39	25	138	8	87		587

Table 3: Confusion matrix for noun supersense tagging. Skipped segments are those absent in training data.

Gold tag	STAT	COMM	COGN	CHAN	EMOT	MOTI	PERC	POSS	SOCI	CONT	skipped	total
v.stative	617	5	21	3	3	14		4	47	1	53	769
v.communic	8	201	3	2		7	1	4	14	4	36	280
v.cognition	14	14	158	1	1	2	11		22	1	25	250
v.change	49	5	2	67		12		6	22	6	39	210
v.emotion	2	7	44	1	72				3	1	12	143
v.motion	5		2	9		77			8		20	122
v.perception		1	16	1	2	1	69		8	1	10	109
v.possession	17	5	1	1		1		43	5		5	79
v.social	16	2	3	2		5			29		18	75
v.contact	10	4	1	2	2	14		4	3	10	15	70
not-a-verb	355	9	9	12	7	7			4			405

Table 4: Confusion matrix for verb supersense tagging. Skipped segments are those absent in training data.

- Adding specific rules for verb-particle constructions, probably based on a lexicon of idiomatic combinations.

- Replacing the CONTIG method by a sequence tagger for contiguous MWEs (e.g. using a CRF), in order to identify unknown MWEs based on generalizations made from known MWEs (Constant and Sigogne, 2011; Schneider et al., 2014).

- Taking parse trees into account to distinguish MWEs from accidental cooccurrences (Nasr et al., 2015).

- Using semantic-based association measures and semantic-based features based on word embeddings to target idiomatic MWEs (Salehi et al., 2015).

- Using fixedness features to identify and disambiguate very productive patterns like ADJ_N (Ramisch et al., 2008; Fazly et al., 2009).

- Developing a more realistic WSD algorithm for supersense tagging, able to tag unseen words and MWEs and to take context into account.

Acknowledgments

This work has been funded by the French Agence Nationale pour la Recherche through projects PARSEME-FR (ANR-14-CERA-0001) and ORFEO (ANR-12-CORP-0005), and by French-Brazilian cooperation projects CAMELEON (CAPES-COFECUB #707/11) and AIM-WEST (FAPERGS-INRIA 1706-2551/13-7). Part of the results presented in this paper were obtained through research on a project titled "Simplificação Textual de Expressões Complexas", sponsored by Samsung Eletrônica da Amazônia Ltda. under the terms of Brazilian federal law No. 8.248/91.

References

Samuel Brody and Mirella Lapata. 2008. Good neighbors make good senses: Exploiting distributional similarity for unsupervised WSD. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 65–72, Manchester, UK, August. Coling 2008 Organizing Committee.

Clara Cabezas, Philip Resnik, and Jessica Stevens. 2001. Supervised sense tagging using support vector machines. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 59–62, Toulouse, France, July. Association for Computational Linguistics.

Matthieu Constant and Anthony Sigogne. 2011. MWU-aware part-of-speech tagging with a CRF model and lexical resources. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 49–56, Portland, Oregon, USA, June. Association for Computational Linguistics.

Silvio Ricardo Cordeiro, Carlos Ramisch, and Aline Villavicencio. 2015. Token-based MWE identification strategies in the mwetoolkit. In *PARSEME's 4th general meeting*.

Mona Diab and Pravin Bhutada. 2009. Verb noun construction MWE token classification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 17–22, Singapore, August. Association for Computational Linguistics.

Afsaneh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, 35(1):61–103.

Kartik Goyal and Eduard Hovy. 2014. Unsupervised word sense induction using distributional statistics. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1302–1310, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Nidhi Kulkarni and Mark Alan Finlayson. 2011. jMWE: A java toolkit for detecting multi-word expressions. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 122–124. Association for Computational Linguistics.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the Fifth International Conference on Systems Documentation (SIGDOC 86)*, pages 24–26, Toronto, Canada.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, (4):553–590, dec.

Alexis Nasr, Carlos Ramisch, José Deulofeu, and André Valli. 2015. Joint dependency parsing and multiword expression tokenization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1116–1126, Beijing, China, July. Association for Computational Linguistics.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69.

Judita Preiss, Ted Briscoe, and Anna Korhonen. 2007. A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 912–919, Prague, Czech Republic, June. Association for Computational Linguistics.

Carlos Ramisch, Paulo Schreiner, Marco Idiart, and Aline Villavicencio. 2008. An Evaluation of Methods for the Extraction of Multiword Expressions. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 50–53, Marrakech, Morocco, June.

Carlos Ramisch. 2015. *Multiword Expressions Acquisition: A Generic and Open Framework*, volume XIV of *Theory and Applications of Natural Language Processing*. Springer. 230 p.

Bahar Salehi, Paul Cook, and Timothy Baldwin. 2015. A word embedding approach to predicting the compositionality of multiword expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 977–983, Denver, Colorado, May–June. Association for Computational Linguistics.

Agata Savary. 2009. Multiflex: A Multilingual Finite-State Tool for Multi-Word Units. In Sebastian Maneth, editor, *CIAA*, volume 5642 of *Lecture Notes in Computer Science*, pages 237–240. Springer.

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

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*, San Diego, California, USA, June.

John Sinclair, editor. 2012. *Collins COBUILD phrasal verbs dictionary*. Harper Collins, Glasgow, UK, third edition. 528 p.

Kiyoko Uchiyama, Timothy Baldwin, and Shun Ishizaki. 2005. Disambiguating japanese compound verbs. *Computer Speech and Language*, 19(4):497 – 512.