Mitigating Silence in Compliance Terminology during Parsing of Utterances

Esme Manandise
Intuit Futures
2600 Marine Way
Mountain View, CA 94043
USA
esme_manandise@intuit.com

Conrad De Peuter
Intuit
2600 Marine Way
Mountain View, CA 94043
USA
Conrad_DePeuter@intuit.com

Abstract

This paper reports on an approach to increase multi-token-term recall in a parsing task. We use a compliance-domain parser to extract, during the process of parsing raw text, terms that are unlisted in the terminology. The parser uses a similarity measure (Generalized Dice Coefficient) between listed terms and unlisted term candidates to (i) determine term status, (ii) serve putative terms to the parser, (iii) decrease parsing complexity by glomming multi-tokens as lexical singletons, and (iv) automatically augment the terminology. We illustrate a small experiment with examples from the tax-and-regulations domain. Bootstrapping the parsing process to detect out-of-vocabulary terms at runtime increases parsing accuracy in addition to producing other benefits to a natural-language-processing pipeline, which translates arithmetic calculations written in English into computer-executable operations.

1 Introduction

The task of extracting multi-token terms [1], i.e. terminological units which denote concepts and entities in a domain, is a core task of Natural Language Processing (NLP). Within the tax-and-regulations domain, some terms are compositional (Nunberg et al., 1994; Baldwin, 2006; Krcmar et al., 2013; Boguraev et al., 2015) [2] in meaning and/or in form, such as *unmarried college student* or *estimated tax payment*; others are mixed instances of compositionality such as *taxable sick leave pay* or *cannabis duty payable*. In addition, terms can correspond either to the canonical form of the concept or to variant forms of concepts' names as in *spouse or common-law partner credit* versus *spouse's or common-law partner's credit*, or *spouse amount* versus *spousal amount* (Park et al., 2002).

From the perspective of parsing raw text, having multi-token terms not only simplifies the input by grouping multi-tokens as singletons but also removes syntactic complexity as the internal structure of these expressions remains opaque to parsing (Korkontzelos et al., 2010; Wehrli, 2014; Boguraev et al., 2015; Nerima et al., 2017). In addition, having multi-token terms allows parsers to output structurally-similar parses for sentences that are constituent-wise similar even if the intra-phrasal complexity of multi-token terms vary.

One major issue in term extraction, known as *silence*, is the failure to extract terms that appear infrequently in a domain-corpus but that domain-specialists would include in a term lexicon. An example of *silence* is when terms such as *inventory valuation* or *combination money purchase* do not make it into the terminology because they occur infrequently in our domain corpus [3]. In the tax-and-regulations domain, the problem with terminology *silence* is particularly acute, as there are many instances of rules and arithmetic calculations concerning a specific entity which is mentioned once in the entire corpus. While

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http:// creativecommons.org/licenses/by/4.0/.

[1] We are not focused on single-token terms. These were extracted in a separate base lexicon.

[2] Compositionality whereby the meaning of an expression is a function of the meaning of its immediate constituents and the syntactic rules used to combine them.

[3] Our corpus consists of a collection of tax forms and instructions for Canada (https://www.canada.ca/en/revenue-agency/services/forms-publications.html) and the United States (https://www.irs.gov/forms-instructions). Approximately 20,000 multi-token terms were automatically extracted. The terms are not vetted by human experts. The parser itself is tasked with choosing the best listed terms for the raw utterance being parsed.

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 204–212
Barcelona, Spain (Online), December 12, 2020.

the entity is not mentioned enough to be highly scored by collocation-based measures during the term extraction process, for the purposes of automatically interpreting and representing tax-and-regulations content, it is essential the entity be treated the same as other items in its class by the parser.

This paper describes a simple method for extracting automatically multi-token terms that are not listed in the compliance terminology at the start of the parsing process of unlabelled utterances. The compliance terminology for taxes and regulations is the result of prior work on identifying, given the domain corpus, concepts and entities by means of co-occurrence/collocation-based surface statistical measures. In addition, linguistic-rule-based filters exclude a set of ill-formed terms from the final list. Because out-of-vocabulary (OOV) terms are tax-and-regulations–specific, we cannot rely on external lexical resources as these expressions are unlikely to be listed in general-purpose, financial or business lexicons. In our approach, the detection of OOV terms takes place during parsing. We use Generalized-Dice-Coefficient-based (GDC) metrics [4] to estimate the degree of similarity between established terms and OOV term candidates. When the GDC-based detection of multi-token terms is enabled during parsing of utterances, experiments show improvements of 93% in parsing accuracy for utterances with OOV terms at the start of the parsing process.

2 Motivation

The goal of the compliance-domain parser is to output, in a simplified logical form (SLF) (Wang et al., 2015, Constant et al., 2016), a semantic representation of utterances taken from tax forms written in English [5]. Of particular interest, are utterances that express entire or partial arithmetic calculations. Downstream components of our NLP pipeline interpret the SLFs and automatically transform them into executable operations.

In our compliance domain, the language of arithmetic calculations written in English is distributed along a continuum of syntactic complexity. Table 1 lists some pairs of utterances and their corresponding SLF (terms are denoted by the underscore) [6].

No.	Calculation Utterances	SLF
1	Enter the sum of exclusion of income from Puerto Rico and form 4563 line 15.	add(exclusion_of_income(puerto_rico),form(4563,line(15)))
2	Enter $1,195 or the total of your employment income you reported on lines 101 and 104 of your return, whichever is less.	min(1195, employment_income(add(line(101),line(104))))
3	Enter $75,300 if married filing jointly or qualifying widow(er)	ifte(or(eq(filing_status(taxpayer),married_filing_jointly), eq(filing_status(taxpayer),qualifying_widow(er))),75300)

Table 1: Calculation utterances and corresponding SLF.

The SLF of utterance 1 is an addition between the amount denoted by the multi-token term *exclusion of income* and the amount on a specific line of a specific form. Utterance 2 is about a choice: the smallest amount of two amounts (*min* operator). The notion of *smallest* is conveyed by a discontinuous dependency as a relative clause at the end of the utterance, namely, *whichever is less*. The first amount for the choice is a dollar constant amount; the second is an addition of the amounts denoted by the term *employment income*, expressed by *total ... on lines 101 and 104*. In utterance 3, inputting the constant

[4]The Generalized Dice coefficient is a similarity measure used in lexicography and term extraction to compute the lexical cohesion of multi-token term candidates. Park et al., 2002 and Kozakov et al., 2004 discuss at length the Dice-Coefficient and the Generalized-Dice-Coefficient statistical measures.

[5]We use a lexical-functional-based parser developed in-house for the compliance domain. In particular, it is tailored to interpret unannotated utterances that express arithmetic calculations written in English

[6]*ifte* in SLFs is a token to denote Boolean branching conditions.

amount of *$75,300* to a calculation is conditional on satisfying one of the disjuncts (*or* operator) for the filing status of the taxpayer, expressed by the multi-token terms *married filing jointly* and *qualifying widow(er)*.

When multi-token terms are missing from the terminology, the task of the parser is to determine the dependencies between the individual tokens that make up the utterances. With more available tokens to parse, the chance of an inaccurate parse increases. Note that, with a large terminology (over 20,000 lexical entries), the prior acquisition of multi-token expressions that are in fact not multi-token domain-concepts is a possibility [7]. If multi-token terms contain pieces of calculations that should be discrete, the parser will not accurately break down the calculation into its constituents, since multi-token terms are monolithic literal strings to the parser–regardless of how many spaces there are between the tokens of a term.

Consider the expressions *married and retired* versus *tuition and fees*. The first expression is not a term; it describes a Boolean operation, where the NLP pipeline must check separately whether the taxpayer is both *married* and *retired*. In contrast, the second expression is a term describing a single entity which is only specific to educational-related forms and instructions. *Tuition and fees* does not occur frequently-enough in our corpus to have made it into the terminology. Extracting, during parsing, the OOV expression *tuition and fees* as a GDC-based term candidate prevents interpreting *tuition and fees* as a Boolean operation. The method identifies the OOV term *tuition and fees* as *similar* to *pensions and annuities*-a term in the existing terminology. Additional examples are provided in Table 2 [8].

GDC Term Candidate	**POS Pattern**	**POS GDC Score**	**Token GDC Score**	**Similar Listed Term**
investor tax credit	[NN, NN, NN]	1.00	0.67	input tax credit
mining exploration tax credit	[NN,NN,NN,NN]	1.00	0.75	mineral exploration tax credit
universal child care	[JJ, NN, NN]	0.67	0.67	specified child care
unused Ontario tuition	[JJ, NNP, NN]	0.80	0.60	unused federal tuition

Table 2: GDC-based term candidates and similarity to terms in terminology.

Finally, the interpretation of utterances with and without OOV terms is reflected in the corresponding SLF. Contrast utterances in Table 3.

No.	**Term**	**Utterances**	**SLF**
1	Yes	amount for an eligible dependant, claim $85.00	ifte(eligible_dependant,85.00)
2	OOV	amount for a qualified dependant, claim $85.00	ifte(dependant(qualified),85.00)
3	Yes	amount for a single parent's qualified dependant, claim $64.00	ifte(single_parent_qualified_dependant,64.00)
4	OOV	amount for a single parent's eligible dependant, claim $64.00	ifte(eligible_dependant(single_parent),64.00)

Table 3: Utterances and SLF with/without terms.

Utterances 1 and 2 differ by the tokens *eligible* versus *qualified* where *eligible* is part of the term *eligible dependant* in utterance 1. While 3 contains the single four-token term *single parent qualified dependant*, utterance 4 counts two separate two-token terms, namely, *single parent* and *eligible dependant*. In the case of utterances 2 and 4 with OOV terms, the parser parses *qualified* and *single parent*

[7] Extraction of invalid term candidates is called *noise*–the opposite of *silence*.

[8] The part-of-speech (POS) names are Penn tags where JJ stands for adjective, NN for singular noun, and NNP for proper noun.

as left modifiers of the head of the noun phrase. In SLFs, the modifiers are enclosed in parentheses as arguments to the predicates, respectively *dependant* in utterance 2 and *eligible_dependant* in utterance 4.

Intuitively, each of the utterances in Table 3 should be of the form *if X, Y* where *X* is a multi-token term with no internal structure for the parser to consume. Otherwise, there is a discrepancy in predicate-argument relations across syntactically- and/or semantically-similar utterances [9].

The SLFs of utterances 2 and 4 of Table 3 should be as in Table 4 below such that their SLFs align with those of utterances 1 and 3 of Table 3 (repeated in Table 4).

No.	Term	Utterances	SLF
1	Yes	amount for an eligible dependant, claim $85.00	ifte(**eligible_dependant**,85.00)
2	Yes	amount for a qualified dependant, claim $85.00	ifte(**qualified_dependant**,85.00)
3	Yes	amount for a single parent's qualified dependant, claim $64.00	ifte(**single_parent_qualified_dependant**,64.00)
4	Yes	amount for a single parent's eligible dependant, claim $64.00	ifte(**single_parent_eligible_dependant**,64.00)

Table 4: SLF unification.

Even with an increase in the size of the domain corpus, there is no guarantee that the OOV terms *qualified dependant* or *single parent eligible dependant* of Table 3 will occur frequently-enough to make it into a term lexicon (as the result of a terminology extraction process)[10]. When processing arithmetic calculations automatically from utterances in English, our compliance system has one shot at outputting SLFs for each calculation such that downstream components can interpret them and transform them into executable operations. SLFs need be accurate. Bootstrapping parsing with a method that detects, on-the-fly, OOV term candidates increases parsing accuracy for these utterances.

3 Experiment

In order to measure the impact of detecting GDC-based terms at runtime on the success of the NLP pipeline that extracts and interprets arithmetic calculations in the tax-and-regulations domain, we ran the following experiment.

3.1 Functional Setup

We used, as a baseline lexical resource, the latest version of the terminology created by our term-extraction process. Separately, we fitted the parser with a preprocessing module to generate multi-token terms from each input utterance. In effect, the preprocessor functions as a chunker with a small grammar to define nominal phrases in English. The grammar sanctions the sequences of adjacent single-token words, which are grouped on the basis of linguistic properties (POS, semantic features, and/or WordNet-based similarity traits); in other words, the grammar rules out non-linguistic *n-gram* word sequences. For instance, for the utterance *Capital gains on gifts of property to qualified donees*, the preprocessor will generate two nominal multi-token term candidates *capital gain* and *qualified donees* [11].

For each term candidate generated by the preprocessor, the process retrieves, from the existing terminology, all terms that share a similar POS pattern. POS patterns need not be identical to allow for complimentary variance like *spouse* versus *spouse's* versus *spousal* in terms such as *spouse amount*, *spouse's amount*, and *spousal amount*. For instance, the POS pattern [NN, NN] is a permissible match with [JJ, NN]. Table 5 lists further matches [12].

To compute the lexical cohesion of the OOV multi-word candidate terms detected during parsing, we

[9] Boguraev et al (2015) make a similar observation about ESG, the parser used in IBM Watson.

[10] Note that, in the examples discussed in Table 3, term extraction did not detect a four-token term with *eligible* but did detect

POS Pattern	Permissible Matches
[NN, NN]	[NN. NN] [NNP, NN] [NNPS, NN] [JJ, NN] [JJR, NN] [JJS, NN] [VBG, NN] [VBN, NN]

Table 5: Permissible POS-pattern matches.

used the GDC statistical measure (Park et al.2002, Kozakov et al. 2004)[13]. The idea was to measure the *termness* of the n-gram candidate terms formed at the onset of parsing with the term-generation preprocessor. Further, the GDC-values could help indicate whether, even weakly-associated n-gram word sequences with at least one identical anchor-token between the candidate and related terms in the terminology, can be extracted as candidate terms to use for parsing of the utterance. We can extract candidate terms whose combined tokens are lexically-cohesive to a certain limit.

The GDC metrics used:

1. all of the terms listed in the terminology that have similar POS patterns.
2. all of the POS patterns from the terms retrieved from the terminology.
3. all of the terms with at least one token shared between a preprocessor-generated term candidate and the terms retrieved from the terminology.
4. on all of the single tokens of the terms retrieved from the terminology.
5. on all of the dictionary values associated with each of the single tokens that make up the terms retrieved from the terminology.

Table 6 below lists some examples of GDC-based terms, scores and closest-related term from the terminology.

When parsing of an utterance completes, GDC-based terms, if any, are added to the terminology. Note that the compliance terminology does not consist merely of a list of terms, but rather the terminology is a dictionary of pairs like [key:values]. The GDC-based terms are added as new entries augmented with a set of default values from established related terms already in the terminology. Adding GDC-based terms to the terminology upon completion of the parsing of an utterance makes the terms immediately available to the utterances that remain, if any, in the parser's queue.

3.2 Steps for Detecting OOV Terms

1. Tokenize utterance.
2. For each single token, do morphological stemming and return base forms of individual tokens.
3. Look up each base form in the single-token base lexicon. Retrieve all lexical data of base form.
4. Generate POS-based term candidates (all legal combinations from left to right) that resolve in a nominal phrase of preset length.

eligible in a two-token term. With *qualified*, it is the reverse–a four-token term but no two-token term.

[11]The preprocessor rules out patterns that include multiple prepositions. For a discussion on term extraction and the nature of terms, see Boguraev et al., 2015.

[12]The Penn tags NN, NNP, NNPS, JJ, JJR, JJS, VBG and VBN correspond to, respectively, singular noun, singular proper noun, plural proper noun, adjective, comparative adjective, superlative adjective, gerund/present participle and past participle.

[13]Park et al., 2002 and Kozakov et al., 2004 discuss at length the statistical measures

GDC Term Candidate	POS Pattern	GDC Score for POS Pattern	GDC Scores for Token Relatedness	Related Listed Term
qualified small business corporation	[VBN,JJ,NN,NN]	1.00	0.25	qualified principal residence indebtedness
total qualified expenditures	[JJ,VBN,NNS]	0.67	0.33	total municipal bonds
qualified expenditures	[VBN, NNS]	0.50	0.50	qualified education
qualified resource	[VBN, NN]	1.00	0.67	qualified education
qualified resource property	[VBN,NN,NN]	1.00	0.33	qualified retirement plan

Table 6: New terms containing *qualified* detected at runtime.

5. For each term candidate generated in step 4, check current baseline terminology for possible matches. If term candidate exists in terminology, remove candidate from term-candidate list generated in step 4.

6. For each term candidate generated in step 4, retrieve all terms that have similar POS patterns. Retrieve the terms as [key:values] pairs.

7. Compute all the GDC metrics as described in subsection 3.1 above.

The terms in the leftmost column of Table 6 are terms produced by our approach.

3.3 Evaluation

The goal of our evaluation was two-fold:

- Determine the number of new multi-token terms extracted on-the-fly during parsing.
- Evaluate the impact of GDC-based terms on parsing accuracy.

In one experiment, we created a corpus of unlabelled utterances collected from tax forms and instructions published in 2017 by the Internal Revenue Service of the United States. We further narrowed the focus to utterances that included the token *qualified*. The test set counted 6,553 utterances with the token *qualified* .

First, we parsed each utterance in the test set only with the latest version of the terminology to create a baseline consisting of utterances paired with their corresponding SLF; the output name is *BTPrun* [14]. Second, we parsed each utterance in the test set with the baseline terminology used in *BTPrun* but with GDC-based-extraction enabled; the output is called *BTGDCrun* [15].

In the BTPrun, 1099 multi-token terms with *qualified* were detected. In contrast, at the end of BTGDCrun, 1,380 terms were extracted– a gain of 281 new multi-token terms extracted during parsing (or 4.3%) (see Table 7).

Finally, we had two human annotators perform a quality assessment of the SLFs with GDC-based terms acquired during parsing. A comparison of the SLFs output by each of the runs yielded 724 changes in the output of the BTGDCrun. The annotation schema was simple: given the utterance, is the new SLF correct? The evaluation was binary: 1 for correct SLF, 0 for ill-formed SLF. Even with a stringent schema for manual evaluation, annotators may judge outcomes differently. However, for this task, judgments were close. Averaging the numbers of SLFs judged as having been corrected through the detection of terms during parsing, we saw a 93% improvement in parsing accuracy for the test set (see Table 8 below).

[14] BTPrun stands for parsing with baseline terminology and baseline parser with **no** GDC-built-in method.

[15] BTGDCrun stands for parsing with baseline terminology and with parser where GDC-based extraction is enabled.

	Total of Utterances in Test Set	Total of *qualified* terms	Percentage of *qualified* terms
BTPrun	6,553	1,099	16.7%
BTGDCrun	6,553	1,380	21%

Table 7: Totals and percentages of terms detected during parsing of utterances in test set.

	Total of Utterances in Test Set	Total of modified SLFs in BTGDCrun	Number of Improved SLFs
	6,553	724	
Annotator 1			669
Annotator 2			675

Table 8: Manual evaluation of modified SLFs in BTGDCrun.

In addition, for each improvement, the annotators were asked to classify the changes in the modified SLFs according to at least one of the following three categories:

- Term matching
- SLF well-formedness
- SLF generation

Consider the three cases of Table 9.

No.	Partial Utterances	SLF in BTPrun	SLF in BTGDCrun
1	qualified reimbursements	reimbursement(qualified)	qualified_reimbursement
2	qualified production activities income	income(qualified_production)	qualified_production_activities_income
3	voluntary employee contributions to a qualified retirement plan (including the federal thrift savings plan)	UNSPECIFIED	voluntary_employee _contribution(and(qualified_retirement _plan, federal_thrift_savings_plan))

Table 9: SLF improvements with GDC-based terms.

In examples 1-3 of the BTGDCrun, the SLFs include novel multi-token terms. The SLF of example 2 is more accurate as *activities* is included as a content-bearing token of the term. In example 3 of the BTGDCrun, the detection of the OOV term *qualified retirement plan* enables the parser to push one possible interpretation for the content in the parenthetical material. Note that, in the BTPrun, the parser did not output any SLF, which is indicated by the convention *UNSPECIFIED*.

Table 10 summarizes the classification by each of the human annotators. Interestingly, the annotators' judgments about the class of improvements by GDC-based terms on SLFs align.

4 Additional Downstream Benefits

An additional advantage for this method occurs further downstream in the NLP pipeline, where elements of parsed phrases are matched to an internal data model. When executing the calculation for *ifte(qualified_dependant,85.00)*, we use a custom-built entity-recognition system to determine the value of *qualified_dependant*. One of the features of this entity recognition system is consecutive token

	Improved SLFs	Term Matching	SLF Well-Formedness	SLF Generation
Annotator 1	669	597	49	23
Annotator 2	675	603	49	23

Table 10: Classification of modified SLFs in BTGDCrun.

matches. As the internal named entity has a description of *IsQualifiedDependant*, having the parser output *ifte(qualified_dependent,85.00)* instead of *ifte(dependent(qualified),85.00)* increases the likelihood of predicting the correct entity.

5 Conclusion

In this paper, we have described a method which increases term recall and improves parsing accuracy of utterances with OOV terms at the start of the parsing process. In addition, multi-token terms detected at parsing runtime are automatically added to the existing terminology. We use a parser fitted with a term-generation preprocessor to identify similarity between OOV multi-token term candidates and multi-token terms listed in the terminology. We observe improvements not only in the interpretation and representation of the utterances by our parser but also in the transformation of the SLFs into executable operations in downstream components of our NLP pipeline.

This method is best suited for domains where precision in a term lexicon, which has been automatically extracted, is important and where the problem with term *silence* can be severe [16] In the future, we would like to experiment with beginning the parsing process by using a terminology manually curated by domain experts. Because the method is domain-agnostic, we do not believe the discovery of OOV terms and augmentation of a domain-specific terminology (as long as there is a preexisting terminology at outset of a parsing task) is constrained to our example domain.

This work has shown that it is possible to overcome term *silence* by adding functionality to a parser with a preprocessor to discover OOV terms on the fly.

Acknowledgements

We gratefully acknowledge the comments by three anonymous reviewers. Not everyone agreed with everything but each comment and suggestion helped the co-authors clarify points in the paper and start a TODO list for future development and testing.

References

Timothy Baldwin. 2006. Compositionality and multiword expressions: Six of one, half a dozen of the other? In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, page 1, Sydney, Australia, July. Association for Computational Linguistics.

Branimir Boguraev, Esme Manandise, and Benjamin Segal. 2015. The bare necessities: Increasing lexical coverage for multi-word domain terms with less lexical data. In *Proceedings of the 11th Workshop on Multiword Expressions*, pages 60–64, Denver, Colorado, June. Association for Computational Linguistics.

Stephen Cohen. 2005. Words! words! words!: Teaching the language of tax. *Journal of legal education*, 55, 12.

Matthieu Constant, Joseph Le Roux, and Nadi Tomeh. 2016. Deep lexical segmentation and syntactic parsing in the easy-first dependency framework. In *Proceedings of NAACL-HLT 2016*, pages 1095–1101. Association for Computational Linguistics, June.

Michael Curtotti and Eric McCreath. 2011. A corpus of Australian contract language: Description, profiling and analysis. pages 199–208, 06.

[16] Our method can introduce *noise*, i.e. wrongly glom a multi-token expression as a single term. However, the individual tokens of a candidate term created by the preprocessor remain available to the parser, which relies on additional corpus-based decision heuristics to rank SLF parses with or without GDC-based terms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Isabella Distinto, Nicola Guarino, and Claudio Masolo. 2013. A well-founded ontological framework for modeling personal income tax. 06.

Ioannis Korkontzelos and Suresh Manandhar. 2010. Can recognising multiword expressions improve shallow parsing? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 636–644, Los Angeles, California, June. Association for Computational Linguistics.

Lev Kozakov, Youngja Park, Tong-Haing Fin, Youssef Drissi, Yurdaer Doganata, and Thomas Cofino. 2004. Glossary extraction and utilization in the information search and delivery system for ibm technical support. In *IBM SYSTEMS JOURNAL*, pages 546–653. IBM.

Lubomír Krcmár, Karel Jezek, and Pavel Pecina. 2013. Determining compositionality of word expressions using word space models. In *MWE@NAACL-HLT*.

Esme Manandise. 2019. Towards unlocking the narrative of the United States income tax forms. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 33–41, Turku, Finland, September. Linköping University Electronic Press.

Luka Nerima, Vasiliki Foufi, and Eric 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, April. Association for Computational Linguistics.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. In Stephen Everson, editor, *Language*, pages 491–538. Cambridge University Press.

Youngja Park, Roy Byrd, and Branimir Boguraev. 2002. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 60–64, Denver, Colorado, January. Association for Computational Linguistics.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. 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 1332–1342, Beijing, China, July. Association for Computational Linguistics.

Eric Wehrli. 2014. The relevance of collocations for parsing. In *Proceedings of the 10th Workshop on Multiword Expressions (MWE)*, pages 26–32, Gothenburg, Sweden, April. Association for Computational Linguistics.