Resources for the Detection of Conventionalized Metaphors in Four Languages

Lori Levin, Teruko Mitamura, Davida Fromm, Brian MacWhinney, Jaime Carbonell, Weston Feely, Robert Frederking, Anatole Gershman, and Carlos Ramirez
Carnegie Mellon University
lsl@cs.cmu.edu, macw@cmu.edu

Abstract

This paper describes a suite of tools for extracting conventionalized metaphors in English, Spanish, Farsi, and Russian. The method depends on three significant resources for each language: a corpus of conventionalized metaphors, a table of conventionalized conceptual metaphors (CCM table), and a set of extraction rules. Conventionalized metaphors are things like *escape from poverty* and *burden of taxation*. For each metaphor, the CCM table contains the metaphorical source domain word (such as *escape*) the target domain word (such as *poverty*) and the grammatical construction in which they can be found. The extraction rules operate on the output of a dependency parser and identify the grammatical configurations (such as a verb with a prepositional phrase complement) that are likely to contain conventional metaphors. We present results on detection rates for conventional metaphors and analysis of the similarity and differences of source domains for conventional metaphors in the four languages.

Keywords: metaphor detection, conventionalized metaphors, metaphor corpus

1. Introduction

This paper describes a tool suite for extracting conventionalized metaphors in English, Spanish, Russian, and Farsi. Conventionalized metaphors are expressions that are known to most people in a culture such as *gap between rich and poor* and *tax burden*. Following Lakoff and Johnson (1980) and Köveces (2002), we analyze these metaphors as evoking a source from which the imagery is drawn (such as WEIGHT), and a target that is described in terms of the source (such as TAXATION). TAXATION IS A WEIGHT is an example of a conceptual metaphor (CM) for which the actual linguistic string might be *heavy taxes*. We use the abbreviation CCM to refer to such conventionalized conceptual metaphors. This paper covers CCMs related to the target domains of poverty, wealth, and taxation, although the methodology is general and can be applied to CCMs in other target domains.

The specific lexical items that are used to express a CCM are called the LMs (linguistic metaphors). Each CCM contains a lexical item from the source domain (such as *burden*), known as the source LM and a lexical item from the target domain (such as *tax*), known as the target LM.

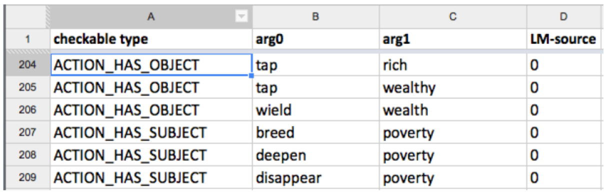
Our tool suite contains three significant resources that may be of general use in other tasks. The first is a corpus of sentences containing CCMs for each language. The second is a table of CCMs in each language including source LM, target LM, and the grammatical construction in which to find them. The third is a set of rules for each language that operate on the output of a dependency parse and detect the grammatical constructions that are relevant for CCM detection.

As our title states, this work focuses only on conventionalized metaphors, and presents specialized methods and resources that enable us to detect them with high precision. Other work on metaphor detection, such as (Gandy et al., 2013) and most of the papers in (E.Shutova et al., 2013), cover novel as well as conventionalized metaphors, often relying on failure of semantic selectional restrictions to indicate metaphor. Our own group has also worked on more general metaphor detection (Tsvetkov et al., 2013; Tsvetkov et al., 2014). It is worth noting that in our experience, the large majority of naturally-occurring metaphors in real corpora are of the conventionalized type, which are detected well using the tools and resources presented here. It is also worth noting that another paper by our group in this conference (MacWhinney and Fromm, 2014) compares specialized and general approaches to metaphor detection. Section 2 describes the CCM tables and the human workflow required to build them, including corpus creation. Section 3 describes the *checkables*, the grammatical constructions in which CCMs are typically found, and the checkable extractor. Section 4 presents results of CCM detection on our test suite. Section 5 presents a comparative analysis of CCMs for the target domain of poverty in English, Spanish, Russian, and Farsi.

2. The CCM Tables

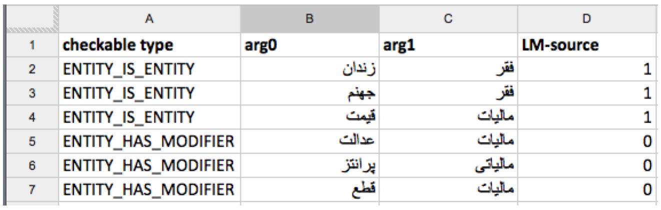
Figures 1 and 2 show rows from the English and Farsi CCM tables. The first column is the checkable relation, which is the grammatical construction in which the source and target LM are found. The first row of Figure 1 represents the the LM *tap the rich* as in the sentence *We want to find new ways to tap the rich.* The first column shows the grammatical construction in which this LM will be found. ACTION_HAS_OBJECT indicates that *tap* and *rich* are a verb and direct object. ACTION_HAS_OBJECT has two arguments an ACTION and an OBJECT. The second and third columns show the lemmas of the two arguments, namely *tap* (`arg0`) and *rich* (`arg1`) in the order in which they are mentioned in the first column. The fourth column indicates which word in the LM is from the source domain. In this case, *tap* (`arg0`) is from the source domain (`FORCEFUL EXTRACTION`).

Other columns not shown here identify the general conceptual metaphor that is exemplified by each linguistic metaphor, such as POVERTY IS CONTAINER for *deepen poverty*. The English CCM table contains 963 CCMs; Russian, 428; Farsi, 911; and Spanish, 866. The CCM tables have been made available in Meta-Share along with another paper by our group (MacWhinney and Fromm, 2014).



	A	B	C	D
1	**checkable type**	**arg0**	**arg1**	**LM-source**
204	ACTION_HAS_OBJECT	tap	rich	0
205	ACTION_HAS_OBJECT	tap	wealthy	0
206	ACTION_HAS_OBJECT	wield	wealth	0
207	ACTION_HAS_SUBJECT	breed	poverty	0
208	ACTION_HAS_SUBJECT	deepen	poverty	0
209	ACTION_HAS_SUBJECT	disappear	poverty	0

Figure 1: Segment of the English CCM Table



	A	B	C	D
1	**checkable type**	**arg0**	**arg1**	**LM-source**
2	ENTITY_IS_ENTITY	زندان	فقر	1
3	ENTITY_IS_ENTITY	جهنم	فقر	1
4	ENTITY_IS_ENTITY	قیمت	مالیات	1
5	ENTITY_HAS_MODIFIER	عدالت	مالیات	0
6	ENTITY_HAS_MODIFIER	پرانتز	مالیاتی	0
7	ENTITY_HAS_MODIFIER	قطع	مالیات	0

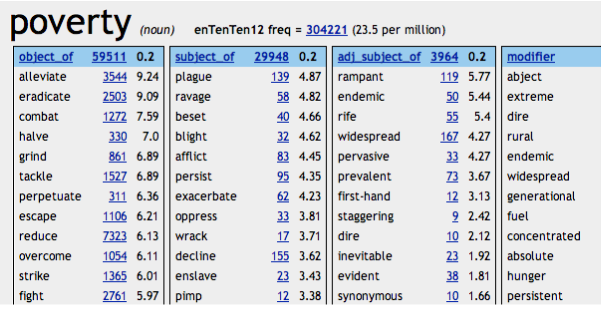
Figure 2: Segment of the Farsi CCM Table

We now describe the human workflow for producing CCM tables. The first step is to use search for sentences containing words such as *poverty* pertaining to the target domain. Most of our searches were conducted using Sketch Engine (sketchengine.co.uk), which produces word sketches as explained below. We searched for words related to poverty, wealth, and taxation in all four languages. SketchEngine provides large corpora for English (11,191,860,036 words), Russian (15,763,181,803 words), and Spanish (8,444,780,226 words), as documented by Jakubicek Jakubicek et al. (2013). We have also contributed to SketchEngine a Farsi corpus that that currently contains 474,733,547 words, which is described in detail in (MacWhinney and Fromm, 2014). All of the words in these large corpora have been lemmatized and the lemmas have been tagged for part of speech.

SketchEngine also provides a method for constructing WordSketches using regular expression grammars that track dependency relations (Ivanova et al., 2008; Khokhlova and Zakharov, 2010). Using these WordSketches, it is easy to track collocations centered around words from the target domain. A fragment of the Word Sketch for English *poverty* from the EnTenTen corpus is shown in Figure 3. A quick glance over even this small fragment of the much larger sketch shows how many of these collocations are metaphorical.

Based on these WordSketches and the links they provide to the original passages, human annotators can quickly extract sentences containing conventionalized metaphors for economic inequality. The number of sentences in our CCM corpora for economic inequality are: English, 7657; Farsi, 6500, Russian, 5632; Spanish 4488. The corpora have been made available in Meta-Share along with another paper by our group (MacWhinney and Fromm, 2014).

We have also constructed a SketchEngine corpus and grammar for Farsi. Construction of these resources for Farsi was necessitated by the fact that the existing SketchEngine corpus for Farsi had only 5,616,550 words. Moreover, these words were not lemmatized or tagged for part of speech, and there was no SketchEngine grammar for Farsi. To address this problem, we built a corpus of 494,733,547 words described by (MacWhinney and Fromm, 2014). To this corpus, we applied the Farsi text pre-processing tools provided by Uppsala University (http://stp.lingfil.uu.se/ mojgan/preper.html), to normalize spacing between Farsi words and their affixes. We then used our own Farsi text normalizer, which removes Arabic and Persian diacritics and normalizes variant forms of the Farsi letter ”ye” to a single unicode representation. Finally, we applied our own Farsi part-of-speech tagger, created by TurboTagger (Martins et al., 2010), which was trained on the part of speech tags in our Persian dependency treebank from Dadegan University (dadegan.ir/en). We then constructed a Farsi SketchEngine grammar. An early version of this grammar using only five grammatical relations succeeded in assigning collocations for 55% of the 11,000 uses of *poverty* in our corpus.



poverty (noun) enTenTen12 freq = 304221 (23.5 per million)

object_of	59511	0.2	subject_of	29948	0.2	adj_subject_of	3964	0.2	modifier
alleviate	3544	9.24	plague	139	4.87	rampant	119	5.77	abject
eradicate	2503	9.09	ravage	58	4.82	endemic	50	5.44	extreme
combat	1272	7.59	beset	40	4.66	rife	55	5.4	dire
halve	330	7.0	blight	32	4.62	widespread	167	4.27	rural
grind	861	6.89	afflict	83	4.45	pervasive	33	4.27	endemic
tackle	1527	6.89	persist	95	4.35	prevalent	73	3.67	widespread
perpetuate	311	6.36	exacerbate	62	4.23	first-hand	12	3.13	generational
escape	1106	6.21	oppress	33	3.81	staggering	9	2.42	fuel
reduce	7323	6.13	wrack	17	3.71	dire	10	2.12	concentrated
overcome	1054	6.11	decline	155	3.62	inevitable	23	1.92	absolute
strike	1365	6.01	enslave	23	3.43	evident	38	1.81	hunger
fight	2761	5.97	pimp	12	3.38	synonymous	10	1.66	persistent

Figure 3: Fragment of the WordSketch for *poverty* the EnTenTen corpus

3. The Checkable Extractors

Checkables are pairs of words in syntactic constructions that are likely to contain CCMs. Our checkable categories are:

Action_Has_Subject:	verb-subject pairs
Action_Has_Object:	verb-object pairs
Action_Has_Modifier:	verb-adverb pairs
Entity_Has_Modifier:	noun-adjective pairs
Entity_Has_Entity:	noun-noun pairs for possession
Entity_Is_Entity:	noun-noun pairs for identity
Entity_Entity:	noun-noun pairs for English noun-noun compounds (for English only)
Entity_Prep_Entity:	noun-preposition-noun triples (for English only)

Our checkable extractors operate on the output of a dependency parse and extract pairs of words that are in a checkable relation. The pairs of words are compared against the CCM tables in order to identify CCMs. The pipeline for CCM detection is as follows: First sentences are parsed with dependency parsers. We are using Turbo Parser (Martins et al., 2010) for English, Spanish, and Farsi. Turbo Parser was trained on the Ancora (http://nlp.lsi.upc.edu/freeling/) treebank for Spanish and on the Dadegan (www.dadegan.ir/en) treebank for

Farsi (Feely et al., 2014). For Russian, we used the AOT Parser (www.aot.ru), which produces slightly deeper semantic dependencies. Next, the checkable extractor produces all pairs of words that are in one of the checkable relationships listed above. Finally, the output of the checkable detector is matched against the CCM table. In order to count as a match, the lemma of the LM source, the lemma of the LM target, and the checkable relation must match.

The Russian checkable extractor consists of 32 rules that transform AOT's deep dependency relations into our checkable relations. The checkable extractors for English, Spanish, and Farsi operate on a dependency tree in CONLL-X format (Màrquez and Klein, 2006). We first convert the CONLL table format into a parenthetical notation as shown in Figure 4 and then use T-Regex (http://nlp.stanford.edu/software/tregex.shtml) to identify the checkable constructions.

T-Regex is designed for phrase structure trees in parenthetical notation where the first element after right parentheses is a syntactic category label such as `NP` or `VP` or a part-of-speech and the elements to its right are the daughter nodes that it dominates. In a dependency tree in parenthetical notation, the first element after right parentheses is a dependency label such as `:obj` or `:vmod` and the elements to its right are its surface form, lemma, part of speech, and character offsets, followed eventually by its dependents. When we use T-Regex operators on parenthesized dependency trees, the semantics of the operators is different. For example, we might use the operator for ”dominates” to find the lemma of a word. We have been successful, however, in using T-Regex on dependency trees in parenthetical notation.

```
"When the poor are able to get jobs,
they forge their own path out of poverty."

((:root "forge" "forge" :vb "41,46"
   (:vmod "When" "when" :wrb "0,4"
    (:sbar "are" "be" :vbp "14,17"
     (:sub "poor" "poor" :jj "9,13"
      (:nmod "the" "the" :dt "5,8"))
     (:prd "able" "able" :jj "18,22"
        (:amod "get" "get" :vb "26,29"
          (:vmod "to" "to" :to "23,25" )
          (:obj "jobs" "job"
            :nns "30,34" )))))
   (:p "," "," :punc "34,35" )
   (:sub "they" "they" :prp "36,40" )
   (:obj "path" "path" :nn "57,61"
     (:nmod "their" "they" :prp\$ "47,52" )
     (:nmod "own" "own" :jj "53,56" ))
   (:vmod "out" "out" :in "62,65"
     (:pmod "of" "of" :in "66,68"
       (:pmod "poverty" "poverty"
           :nn "69,76" )))
   (:p "." "." :punc "76,77" )))
```

Figure 4: CONLL format represented with parentheses

Extraction of checkables may seem straightforward because relations like a subject or object dependent of a verb are explicitly represented in a dependency tree. However, the checkable extractor handles three special cases:

Coordinate structures: For a phrase like *escape poverty and despair* our parsers do not produce an `:obj` relation between the verb and each conjunct (because of the treebanks they were trained on). The checkable extractor processes the coordinate structure to produce verb-object pairs ACTION_HAS_OBJECT:`(escape, poverty)` and ACTION_HAS_OBJECT:`(escape, despair).`

Light nouns: Many noun phrases are headed by words like *percent*, *kind*, and *sort*. In a phrase like *identified eleven kinds of insomnia* the parse tree shows an `:obj` dependency between *identified* and *kinds*. The checkable extractor recognizes a list of such light nouns such as quantifiers, partitives, and containers and adds an `:obj` link between the verb and the complement of the light noun to make checkables like ACTION_HAS_OBJECT:`(identify, insomnia).`

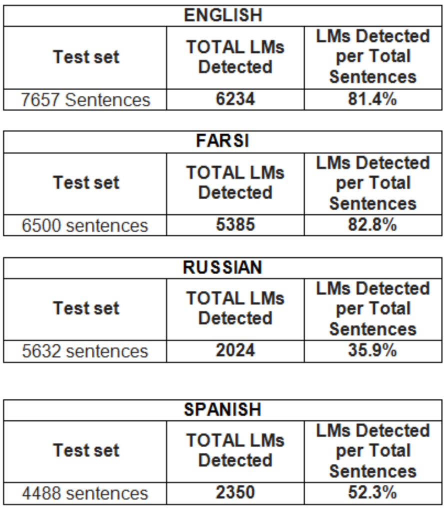
***Toy gun* noun phrases:** *Gun* is syntactically the head of *toy gun*, but semantically, it may be more appropriate to characterize it as a toy. We invoke the *toy gun* rule for phrases like *poverty trap*, *tax burden*, and *cycles of poverty*. For the input *escape the poverty trap*, the checkable extractor produces an `:obj` dependency between *escape* and *poverty* even though *poverty* is not the head of the noun phrase *poverty trap*. In order to detect *toy gun* noun phrases, we have a list of salient words like *poverty* and *taxation* that we look for in modifier position. Note that this is different from light noun detection, where we look for certain non-salient words in head position. We only want *trap* and *cycle* to be ignored when they are modified by specific words like *poverty* and *tax*. For example, we do not want to ignore *trap* in *escape the bear trap*, which is about escaping a trap, not escaping a bear.

4. Results of CCM detection

We tested our CCM detection system on our CCM corpora. In the table below, we report for each language, the recall of the CCM detection system when applied to the CCM corpus. When we are not able to detect CCMs, it is often because of incorrect parses that lead to incorrect checkables. The higher recall rates for English and Farsi as compared to Russian and Spanish reflect project personnel effort and are not indicative of difficulty of the languages or deficiencies of the NLP tools.

5. Crosslinguistic Comparison

In addition to using the CCM corpora as development sets for automated metaphor detection, we used them for cultural analysis of common metaphorical source domains. Because our corpora for the four languages are not comparable in genre or time period, they cannot be directly compared. However, it is interesting to note differences in the frequency of metaphorical source domains. Figure 6 shows examples from our Spanish, Russian, and Farsi corpora for the target domain of poverty. The examples reflect the most frequent source domain for each corpus: DISEASE for English, SCHISM for Spanish, ABYSS for Russian, and LOCATION for Farsi.



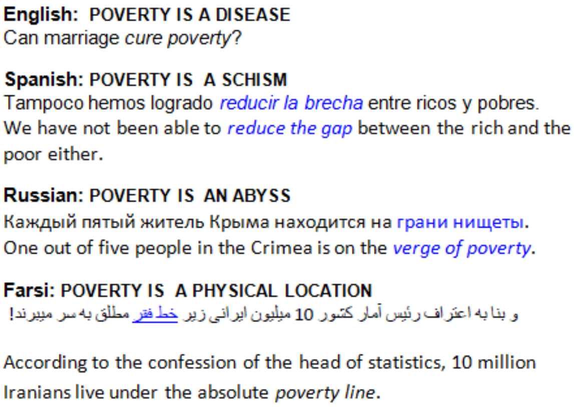
ENGLISH		
Test set	TOTAL LMs Detected	LMs Detected per Total Sentences
7657 Sentences	6234	81.4%

FARSI		
Test set	TOTAL LMs Detected	LMs Detected per Total Sentences
6500 sentences	5385	82.8%

RUSSIAN		
Test set	TOTAL LMs Detected	LMs Detected per Total Sentences
5632 sentences	2024	35.9%

SPANISH		
Test set	TOTAL LMs Detected	LMs Detected per Total Sentences
4488 sentences	2350	52.3%

Figure 5: Results of CCM Detection



English: POVERTY IS A DISEASE
Can marriage *cure poverty*?

Spanish: POVERTY IS A SCHISM
Tampoco hemos logrado *reducir la brecha* entre ricos y pobres.
We have not been able to *reduce the gap* between the rich and the poor either.

Russian: POVERTY IS AN ABYSS
Каждый пятый житель Крыма находится на грани нищеты.
One out of five people in the Crimea is on the *verge of poverty*.

Farsi: POVERTY IS A PHYSICAL LOCATION
و بنا به اعتراف رئیس آمار کشور 10 میلیون ایرانی زیر خط فقر مطلق به سر میبرند!

According to the confession of the head of statistics, 10 million Iranians live under the absolute *poverty line*.

Figure 6: Examples of CCMs in Four Languages

In the target domain of taxation, the most frequent source domain in our English CCM corpus is CRIME, with a majority of TAXATION IS CRIME metaphors using the linguistic metaphor (LM-source) *punitive*. In contrast, in the Russian, Spanish, and Farsi CCM corpora, the most frequent source domain is PHYSICAL BURDEN, using LM source words indicating relief and weight.

6. Discussion

This paper has presented a set of rules and hand-crafted resources for the detection of conventionalized metaphors, which as mentioned in the introduction can comprise the large majority of the metaphors in actual corpora. It is worth noting that research in language technologies often does not distinguish problems that are easily captured by rule- and lexicon-based systems from problems that require advances in machine learning, which also often require very large computational resources to process. But often the largest portion of a problem, in our case metaphor detection, can be solved quickly with relatively simple hand-crafted components, allowing practical applications to move forward while the community continues to research solutions for the harder components of the problem.

Acknowledgements

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C- 0020. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

7. References

E.Shutova, Klebanov, B. B., Tetreault, J., and Kozareva, Z., editors. (2013). *Proceedings of the First Workshop on Metaphor in NLP*. Association for Computational Linguistics.

Feely, W., Manshadi, M., Frederking, R., and Levin, L. (2014). The CMU METAL Farsi NLP approach. In *LREC*.

Gandy, A., Nadji, A., Atallah, M., Friede, O., Howard, N., Kanareykin, S., Koppel, M., Last, M., Neuman, Y., and Argamon, S. (2013). Automatic identifcation of conceptual metaphors with limited knowledge. In *Proceedings of AAAI*.

Ivanova, K., Heid, U., im Walde, S. S., Kilgariff, A., and Pomikalek, J. (2008). Evaluating a german sketch grammar: a case study on noun phrase case. In *LREC*.

Jakubicek, M., Kilgariff, A., Kovar, V., and Rychly, P. (2013). The tenten corpus family. In *Paper presented a the International Conference on Corpus Linguistics*, Lancaster.

Khokhlova, M. and Zakharov, V. (2010). Studying word sketches for russian. In *LREC*.

Köveces, Z. (2002). *Metaphor: a practical introduction*. Oxford University Press.

Lakoff, G. and Johnson, M. (1980). *Metaphors we Live by*. Chicago University Press.

MacWhinney, B. and Fromm, D. (2014). Two approaches to metaphor detection. In *LREC*.

Màrquez, L. and Klein, D., editors. (2006). *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*. Association for Computational Linguistics, New York City, June.

Martins, A., Smith, N., Xing, E., Aguiar, P., and Figueiredo, M. (2010). Turbo parsers: dependency parsing by approximate variational inference. In *EMNLP*.

Tsvetkov, Y., Mukomel, E., and Gershman, A. (2013). Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*. Association for Computational Linguistics.

Tsvetkov, Y., Boystov, L., Gershman, A., Nyberg, E., and Dyer, C. (2014). Metaphor detection with cross-lingual model transfer. In *Proceedings of the Annual Conference of the Association for Computational Linguistics*.