Grasping Metaphors:
Lexical Semantics in Metaphor Analysis

Enrico Mensa, Aureliano Porporato, and Daniele P. Radicioni (✉)

Dipartimento di Informatica – Università degli Studi di Torino, Italy
{mensa,radicion}@di.unito.it,
aureliano.porporato@edu.unito.it

Abstract. Metaphors represent to date an extraordinary challenge for computational linguistics. Dealing with metaphors has relevant consequences on our ability to build agents and systems that understand Natural Language and text documents: annotating metaphoric constructions by linking the metaphor elements to existing resources is a crucial step to make text documents more easily accessible by machines. Our approach tackles metaphors by considering concepts and their abstractness. We report the encouraging results obtained in a preliminary experimentation; we elaborate on present limitations, and individuate the needed improvements, which will be the base for future work.

Keywords: Metaphors, Figurative Language, Lexical Semantics, Natural Language semantics, Lexical Resources.

1 Introduction

Metaphors can be seen as mechanisms for delivering semantic content in a concise way. More concise than with literal, plain language. Metaphors are highly pervasive in both language and thought [5], to such an extent that more than 30% sentences in the British National Corpus contain a metaphor [9]. It is thus easy to foresee that the capacity of recognizing and elaborating metaphorical uses will become more and more essential for building computational systems to deal with Natural Language, such as, e.g., conversational agents underlying chat bots, automatic summarization systems, and in general for extracting information from text documents and in making text documents machine readable. A metaphor is a mechanism to quickly deliver some information: some abstract concept (*explanandum*) is explained by referring to something else (*explanans*), which is more directly understood. Typically, the latter element comes from a more direct physical experience of the real world. The *abstractness* of involved concepts seems to play a major role in metaphors creation and understanding. This work investigates the interplay between metaphor detection and the lexical semantics underlying the terms involved in the metaphorical construction. Our approach consists of two main steps: *i)* we extend the conceptual representation contained in COVER—a recently proposed lexical resource [6]— by automatically annotating information on concepts' abstractness, and *ii)* we propose an

algorithm to detect metaphors and investigate whether some kinds of metaphors can be understood based on abstractness. In particular, we are presently concerned with illustrating how metaphors are identified and mapped onto concept identifiers, which is a relevant step towards the semantic annotation of text documents and the encoding of meaningful pieces of information in machine readable format, such as RDF triples, linked to encyclopedic resources such as DBpedia.

2 Metaphors Detection

The COVER lexical resource was originally conceived as part of a larger project aimed at combining ontological inference and common-sense reasoning [7]. COVER has been built by merging BabelNet [8] and ConceptNet [1], and is composed by a list of vectors, each reporting information about a single concept. The representation of concepts rather than just terms requires the adoption of a set of concept identifiers (so to define a uniform naming space), and COVER relies on the sense inventory provided by BabelNet. BabelNet is a *semantic network* where each node (called synset, that is 'set of synonyms') represents a unique meaning, identified through a BabelNet synset ID (e.g., BN:00008010N). Furthermore, most BabelNet synset IDs are directly linked to the corresponding DBpedia URIs via the EXTERNAL LINKS relation in BabelNet: this connects COVER with the Semantic Web. The conceptual information borrowed from BabelNet has been coupled to common-sense knowledge, that has been extracted from ConceptNet. The ConceptNet relationships have been set as the skeleton of the vectors in COVER, that is the set dimensions upon which a vector describes the represented concept. More precisely, each vector dimension contains a set of values that are concepts themselves, identified through their own BabelNet synset IDs.

We extended the conceptual representation in COVER by enriching each concept herein with information on its *abstractness*[1] [2]; due to the lack of space, we defer to a future work the description of how this annotation was automatically performed. We focus instead on showing how abstractness information is used by the algorithm for the metaphors detection.

2.1 Metaphor detection algorithm

Provided that different categorizations of metaphors can be drawn, we refer to the threefold (not exhaustive) categorization of metaphors proposed in [3]. In this view, *Type I* metaphors are in the form "*smb/sth is sth*" (e.g., "He is a monster"), in which something or somebody is said to be of a kind that is not correct in a literal sense; *Type II* metaphors are in the form "*smb/sth verb sth*" (e.g., "I shot down all his arguments"), where an action is performed by or on something that cannot properly perform an action of that sort; *Type III* metaphors are in the form "*adj noun*" (e.g., "A brilliant idea"), where an adjective is associated to a concept that cannot have the quality expressed in a literal sense. We focus

---

[1] The enriched resource can be downloaded at `http://ls.di.unito.it`.

on metaphors of Types I and II, and presently disregard those of any different type.

Given a sentence $\mathcal{S}$ along with its parse tree $\triangle(\mathcal{S})$, we individuate the dependency patterns corresponding to Type I and II metaphors, which we denote as $\overline{\wedge}(\mathcal{S}) \subset \triangle(\mathcal{S})$. We note that Type I metaphors have a direct counterpart in terms of RDF triples (e.g., "The baby is a new arrival" is represented as IsA(BABY,NEW ARRIVAL)). Conversely, in the case of Type II some further effort is needed in order *i)* to map the verb onto some predicate; and *ii)* to split verbal subcategorization frames with more than two dependents into an appropriate set of triples, according to the semantics of each verb (e.g., "Laughter filled the room in few moments." is represented as FILL(LAUGHTER, ROOM) and FILL(LAUGHTER, [IN] MOMENTS)). However, in all cases the nouns involved in such triples can be mapped onto DBpedia nodes by means of their identifiers.

Some preprocessing steps are performed, basically involving syntactic parsing and word sense disambiguation. Namely, given in input the sentence $\mathcal{S} = \{t_1, t_2, \ldots, t_n\}$ composed of n input terms, we parse it and obtain the parse tree $\triangle(\mathcal{S})$; we then perform the word sense disambiguation of the terms in $\mathcal{S}$,[2] thus obtaining the set of concepts $\mathcal{C}(\mathcal{S}) = \bigcup_{i=1}^{n} \mathrm{WSD}(t_i)$.

The metaphor detection algorithm consists of the following steps:

1. Given the dependency patterns $\overline{\wedge}(\mathcal{S}) \subset \triangle(\mathcal{S})$ on the parse tree, we retain the corresponding concepts (thus dropping patterns whose elements were not disambiguated), $\mathcal{C}' = \bigcup_{\mathcal{C}} \overline{\wedge}(\mathcal{S})$; among concepts $c' \in \mathcal{C}'$, we select *target* (that is, *subj* in both Type I and II metaphors) and *source* (*dir-obj* in Type I metaphors, and *verb* in Type II metaphors) of the metaphorical expression;
2. We label as metaphorical a sentence if the target concept is more abstract than the source concept.

3   Pilot Experimentation

The aim of this experimentation is to test in how far combining syntactic, conceptual and abstractness information can be helpful in unveiling the presence of metaphors. We experimented on the Master Metaphors List (MML), a set of metaphors compiled by Lakoff and others in the '80s [4]. This corpus contains 1728 sentences, each sentence with at least one metaphor. From this set we extracted 75 sentences: we selected 40 sentences containing a metaphor of Type I, and 35 with a metaphor of Type II. We then collected 75 additional non metaphoric sentences (so to be able to compute the *precision* metrics, too); syntactic constructions similar to those characterizing sentences with Type I and Type II metaphors were preserved. The final data set is available at the URL `ls.di.unito.it`.

The system obtained a Recall of 0.70 and 0.74 on Type I and Type II, respectively, and a Precision of 0.56 (Type I) and 0.77 (Type II). The higher accuracy on Type II metaphors corroborates our hypothesis, thereby showing that

---

[2] We presently used Babelfy, `http://babelfy.org` for the WSD and the Stanford CoreNLP, `https://goo.gl/yxcRPF` as our parser.

for such (simpler) cases the comparison between target and source abstractness works fine. An explanation for the lower figures on Type I may stem from the fact that some Type I metaphors require *projecting* some features from the source onto the target (e.g., *lawyers are sharks*). In such cases, we conjecture that just considering the abstractness of the involved terms does not suffice, since the metaphor is best recognized by projecting the features of ferocity and danger-ousness —which is proper to sharks— onto lawyers, as well. Remarkably, these are typically common-sense traits.

4   Conclusions

The experimental results seem to support the proposed approach, that puts together deep parsing, word sense disambiguation, common-sense knowledge and abstractness information. However, it also emerged that such approach needs further, substantial, efforts in order to deal with the widely varied linguistic constructions actually underlying metaphoric language. Grasping the semantics hidden in metaphors may be seen as the task of making explicit how abstract concepts, actions and properties can be explained through less abstract entities, or by resorting to common-sense traits that are transferred from the source to the target. The issue of explanation comes, in other words, together with the detection itself. Improving the whole system and providing it with explanatory skills will be the focus of our future work.

References

1. Havasi, C., Speer, R., Alonso, J.: ConceptNet: A lexical resource for common sense knowledge. Selected papers from RANLP 309, 269–280 (2007)
2. Iliev, R., Axelrod, R.: The paradox of abstraction: Precision versus concreteness. Journal of psycholinguistic research 46(3), 715–729 (2017)
3. Krishnakumaran, S., Zhu, X.: Hunting elusive metaphors using lexical resources. In: Procs of the Workshop on Computational Approaches to Figurative Language. pp. 13–20 (2007)
4. Lakoff, G., Espenson, J., Schwartz, A.: The master metaphor list. Tech. rep., University of California at Berkeley (October 1991)
5. Lakoff, G., Johnson, M.: Metaphors we live by. The Univ of Chicago Press (2003)
6. Lieto, A., Mensa, E., Radicioni, D.P.: A Resource-Driven Approach for Anchoring Linguistic Resources to Conceptual Spaces. In: Procs. of AI*IA. pp. 435–449. LNAI, Springer (2016)
7. Lieto, A., Radicioni, D.P., Rho, V.: Dual PECCS: A Cognitive System for Conceptual Representation and Categorization. JETAI 29(2), 433–452 (2017)
8. Navigli, R., Ponzetto, S.P.: BabelNet: Building a very large multilingual semantic network. In: Procs of the 48th ACL. pp. 216–225. ACL (2010)
9. Shutova, E.: Design and evaluation of metaphor processing systems. Computational Linguistics 41(4), 579–623 (2015)