Using Semantically Annotated Corpora to Build Collocation Resources

M. Alonso Ramos[1], O. Rambow[2] and L. Wanner[3]
[1]Universidade da Coruña, La Coruña, Spain
[2]Columbia University, New York
[3]ICREA and Universitat Pompeu Fabra, Barcelona
E-mail: lxalonso@udc.es, rambow@ccls.columbia.edu, leo.wanner@upf.edu

Abstract

We present an experiment in extracting collocations from the FrameNet corpus, specifically, support verbs such as *direct* in *Environmentalists directed strong criticism at world leaders*. Support verbs do not contribute meaning of their own and the meaning of the construction is provided by the noun; the recognition of support verbs is thus useful in text understanding. Having access to a list of support verbs is also useful in applications that can benefit from paraphrasing, such as generation (where paraphrasing can provide variety). This paper starts with a brief presentation of the notion of lexical function in Meaning-Text Theory, where they fall under the notion of lexical function, and then discusses how relevant information is encoded in the FrameNet corpus. We describe the resource extracted from the FrameNet corpus.

1. Introduction

Collocations, i.e., lexically restricted binary word cooccurrences, pose a challenge for many NLP applications. No wonder that the compilation of collocation resources from corpora has been a popular research topic for over twenty years now.[1] Usually, the corpus is required to be POS- or syntax-annotated. However, POS or syntactic tree annotation is not sufficient to automatically compile and structure collocation resources in the manner of modern collocation dictionaries (*Oxford Collocations Dictionary*, *BBI*, *LTP* etc.), namely, according to semantic (rather than only syntactic) criteria. For this purpose, a corpus annotated with lexical semantic information is crucial. We present an experimental compilation of collocation resources from such a corpus. Our working scenario is as follows:

A. Following the common lexicographic tradition (Cowie, 1993) we assume that a collocation is a restricted binary co-occurrence of lexical units (LUs) between which a syntactic relation holds, and that one of the LUs (the *base*) keeps the semantics it has in isolation, while the semantics of the other (the *collocate*) is predetermined by the combination as a whole. The semantics of collocates can be generalized across all collocation occurrences, allowing to assign to each collocation a semantic class label.
B. As collocation typology, we use *lexical functions* (LFs) from *Explanatory Combinatorial Lexicology* (Mel'čuk, et al., 1995). The LF-typology is arguably the most fine-grained semantic collocation typology. Furthermore, as LFs are relations between LUs and do not reference other semantic notions (such as specific ontologies or meaning theories), they are particularly easy to use in NLP.
C. As corpus, we use the *FrameNet*'s (FN) corpus of examples (Ruppenhofer et al., 2006). This corpus contains both syntactic and semantic annotations, and, very important, seeds of collocation annotation.[2]

We present some experiments on the identification of LFs in the FN-corpus. The experiments demonstrate that it is feasible and beneficiary to use semantically annotated corpora for the compilation of collocation resources.

2. LFs as a semantic collocation typology

LFs are a means to encode restricted lexical co-occurrence and lexico-semantic derivation. More specifically, each LF **f** is an abstract directed lexico-semantic relation that holds between an LU L_k, the *keyword* of **f**, and an LU L_{vi}, the *value* of **f**:[3]

SPEECH	– $\mathbf{f_1}$ –	DELIVER
BUY	– $\mathbf{f_4}$ –	SELL
SPEECH	– $\mathbf{f_1}$ –	GIVE
GIVE	– $\mathbf{f_4}$ –	RECEIVE
WALK	– $\mathbf{f_1}$ –	TAKE
CURE	– $\mathbf{f_5}$ –	DOCTOR
EXAM	– $\mathbf{f_2}$ –	PASS
LETTER	– $\mathbf{f_6}$ –	ADDRESSEE
PROMISE	– $\mathbf{f_2}$ –	KEEP
CRY	– $\mathbf{f_7}$ –	TEARFUL
ARGUMENT	– $\mathbf{f_3}$ –	STRONG
BEND	– $\mathbf{f_7}$ –	FLEXIBLE
SMOKER	– $\mathbf{f_3}$ –	HEAVY
SAND	– $\mathbf{f_8}$ –	GRAIN[4]

[1] Cf. Computer *Speech & Language,* 19(4) for an overview.

[2] Publications on FN (Fillmore et al., 2001; Ruppenhofer et al., 2006) state that it is planned to enrich the FN-corpus by LFs. Our work is a contribution to this endeavour.

[3] For a general presentation of LFs, see (Mel'čuk, 1996); for an interpretation of LFs as a classification typology, see (Wanner, 2004).

[4] As the illustration shows, the same relation can hold between one given keyword LU and several value LUs. In the functional notation (which is irrelevant for our purpose, but which is commonly used in the literature), this reads as: $\mathbf{f}: V \rightarrow \mathbf{P}^V$ or $\mathbf{f}(L_k)$

In total, the LF typology distinguishes about 60 different LFs; each is identified by a Latin abbreviation: f_1 from above is referred to as "$Oper_1$", f_2 as $Real_1$, f_3 as "Magn", f_4 as "$Conv_{31}$", etc.

We can distinguish *paradigmatic LFs* from *syntagmatic LFs*. Paradigmatic LFs represent lexico-semantic derivations (e.g., Conv31 captures the first-third argument conversion, S0 the name of a deverbal noun, Si the name of the *i*-th argument of a predicative LU, etc.). Syntagmatic LFs represent collocations: an intensifying modifier (Magn), a light (=support) verb that takes the *i*th syntactic argument of *Lk* as subject and *Lk* itself as direct object (Operi); etc. The subscripts of the syntagmatic LFs indicate how the argument structure of *Lk* is mapped onto the argument structure of *Lvi*: in Oper1, the first syntactic argument of *Lk* (the keyword) is realized as subject of *Lvi* (the value); in Oper2, the second argument of the keyword functions as the subject of the verb, etc.

3. The basics of the FN-Corpus annotation

FN is a lexical resource for English that records the semantic and syntactic valences of each lexeme in terms of frame semantics (Ruppenhofer et al., 2006). The resource consists of a semantic frame dictionary and an accompanying corpus of frame-annotated sentences. A *frame* is a representation of an abstract situation pattern. It is defined drawing upon its *core frame elements* (FEs) such as Communicator, Experiencer, Evaluee, etc.; cf. the definition of the frame Judgment-Communication (the core FEs are highlighted):

> "A Communicator communicates a judgment of an Evaluee to an Addressee [...] This frame does not contain words where Evaluee and the Addressee are necessarily the same."

In addition to core FEs, a frame may have *peripheral FEs* (e.g., Degree) and *extra-thematic FEs*.

Each frame is associated with a series of LUs that "evoke" it. For instance, Judgment-Communication is evoked by *acclaim*.v, *accusation*.n, *accuse*.v, *belittle*.v, *belittling*.n, *blame*.v, *criticism*.n, etc. Each evoking LU receives a lexical entry with the pointer to the frame and a set of annotated sentences in the corpus. The sentences are annotated with respect to the instantiation of FEs and the governors of the LU that evoke the frame (identified as *target*, Tgt); cf. an annotated sentence associated with *criticism*:

> [<Communicator> Environmentalists] {directed^{Supp} [<Degree> strong] *criticism* ^{Tgt} [<Evaluee> at world leaders] after only 15 minutes was spent discussing the environment at the Group of Seven conference in London

The FE-annotation consists of triples: <FE, *phrase type*, *grammatical function*>, such as <Addressee-NP-Ext> above (Ext stands for *external argument*).

The governor annotation consists of one of the following labels: "Supp", "Ctrlr", "Gov-X" (governor-X). "Supp" is used to tag a support verb governor of an event-denoting noun that serves mainly to project a clause centred on the frame of the noun. Ctrlr (in analogy to syntactic control constructions) tags a verb which introduces an event different from the one evoked by Tgt, and both events share an FE. Thus, the FE "offerer" in the frame evoked by *offer* as in *to offer help* corresponds to the FE "helper" in the frame ASSISTANCE evoked by *help*.

The use of the Gov-X label is twofold. Firstly, Gov marks predicates which are semantically related to the qualia structure of the target artefact noun they govern, marked by X. Thus, *stab* is treated as Governor of *knife*, *fire* as Governor of *weapon*, etc. Secondly, Gov marks predicates which syntactically govern "transparent nouns" and semantically the nouns which syntactically depend on the former. In the example below, *bought* syntactically governs the transparent noun *bunch* and semantically *of red carnations*:

> I had [bought]^{Gov} {a [<Aggregate> *bunch*^{Tgt}] [<Individuals> of red carnations]}^{X}

Also useful for our purposes is the "rcoll" (right collocate) and "lcoll" (left collocate) annotation in the Annotation Report of some LUs. Although to our knowledge not explained in the literature, the label seems to indicate a high probability of co-occurrence between Tgt and the right/left collocate. Note that there is not necessarily a syntactic relation between Tgt and the collocate. Cf. an example of an rcoll-*instruction* in the entry for *follow*:

> I have done as a Hahnemann requested and have FOLLOWED his instructions exactly .

This means that *instruction* frequently occurs in the corpus to the right of *follow*.

4. Indicators of LFs in the FN-corpus

The FN-Corpus reveals five different indicators of LF-instances. Three of them are tags which provide supplementary information to the valency of the LU labelled as Tgt: Supp, Ctrlr, and Governor; the fourth is the peripheral FE Degree, and the fifth is the ''rcoll''/''lcoll'' annotation.

Supp is the most explicit collocation marker and a very strong pointer to Oper-LF instances. However, Supp does not provide any information concerning the syntactic structure, i.e., the distinction between $Oper_1$/$Oper_2$/$Oper_3$. This distinction is made using the FE-distribution (see also below).

Ctrlr is a pointer to semantically loaded collocations: the base evokes a frame with elements shared by the frame evoked by the collocate. This excludes the Oper-type of LFs, subsuming a whole range of semantically loaded LFs such as CausFunc (see (4)), Real (see (5)), whose further differentiation can be subject of further semi-automatic

---

= $\{L_{v1}, L_{v2}, \ldots, L_{vn}\}$. The elements Lvi are approximately synonymous to each other in the context of this relation. Their semantics is specific to the **f**.

classification.

(4) [<Source>These mills] have since provided[Ctrlr] a *source*[Tgt] [<Theme>for material, inspiration, fabrication and construction].

(5) Ever since [<Participant_1>I] won[Ctrlr] a short-story [<Competition>*competition*[Tgt]]

In accordance with the twofold use of the Governor-X label, the LFs that can be deduced and the indications for their deduction are different. In connection with the qualia structure, it hints either at Real (when expressing the Telic role, see (6)) or at CausFunc (when expressing the Agentive role, see (7)). In connection with transparent nouns, it hints at either Mult (see (8)) or Sing.

(6) We have **dropped**[Gov] a cluster BOMB on Carlos Cardoen.

(7) Roads were being levelled, armies raised, {*castles*[Tgt]}[X] erected[Gov]

(8). In winter {[<Aggregate>*colonies*[Tgt]] [<Individuals>of seals]}[X] arrive][Gov] from further north to have their pups.

Degree is a clear indicator of the instances of the Magn- and AntiMagn-LFs (see the following examples).

(9) [<Stimulus>He] *reminded*[Tgt] [<Cognizer>Riven] [<Degree>a little] [<Phenomenon>of the Bicker he had known at the bothy] .

(10) [<Type>This reasoning] {seems}[Supp] [<Degree>slightly] *artificial*[Tgt]

In the case of 'r/lcoll', the keyword of the LF is the LU tagged in the Annotation Report of Tgt as 'r/lcoll' (for example, *instruction*); Tgt is the value of the LF (for example, *follow*); and the LF-label is derived from the Tgt- r/lcoll constellation and the frame. For the frame Compliance evoked by *follow*, the LF is Real.

5. Automatic detection of LFs

Exploiting the indicators of collocational information in the FN-corpus, we can extract or tag LF-instances. Here, we focus on the identification of $Oper_i$-instances using exclusively the Supp-label.

Our basic algorithm for the extraction of $Oper_i$-instances is rather straightforward. It exploits the distribution of FEs and the projection of the semantic to syntactic valency of the individual LFs, using only the annotations available in the corpus. We take the verb tagged as Supp to be the support verb of an $Oper_i$-LF, whose base is Tgt. For the choice of the subscript of Oper, we initially used the following two-part heuristic:

1. If the FE of the subject of the Supp is Agent, Person, Speaker, or Helper, we assume that this FE is the first argument and thus the subject of the underlying verb, and choose '1' as subscript.
2. If the FE of the subject of the Supp is something else, we use the ordering of the core FEs given in the frame of the base noun as a heuristic for the syntactic subcategorization. Thus, we verify which FE is the subject, and then choose as subscript for the Oper the position of that FE in the list of core FEs.

Cf. examples for 1 and 2:

(11) *With reluctance, Morton decided that* [FE= AGENT *he*] *must* [Supp *make*] *another* [Tgt *attempt*] *to identify the dead girl* (Agent is subject of Supp; therefore the hypothesized LF is $Oper_1$)

(12) *Yet the* [FE= EVALUEE *Franks*] [Supp *have received*] [Tgt *criticism*] [Reason *for including a lot of songs dedicated fans will already own*] (Evaluee (= the second FE) is subject of Supp; therefore, the hypothesized LF is $Oper_2$).

However, we subsequently recognized that only considering the core FEs is too limiting, since often non-core FEs appear as subjects of support verbs. But if we consider all FEs, it is almost impossible to recreate the MTT convention for numbering the arguments of the nominal predicate. This is not surprising as this is a lexicographic convention which is not derivable from the conventions of a different formalization of lexical semantics, namely FrameNet. We therefore decided to abandon a literal implementation of the MTT nomenclature, and instead identify the type of Oper support verb by the FE of the subject of the support verb, for example: $Oper_{Agent}$, $Oper_{Evaluee}$.

To use this information in paraphrasing (i.e., deriving *We assist the protection of this heritage* from *We provide assistance for the protection of this heritage*), we would need on additional piece of information: we first need to find the verb related to the keyword noun, and then determine how the FEs of the verb are mapped to syntactic arguments of the corresponding noun. These steps are possible using the resources in FrameNet, but we do not report on them here.

6. Results

We ran our evaluation on all verbs in the 1.3 release of FrameNet. After eliminating cases other than nouns with support verbs (Supp is also used to annotate other part-of-speech pairs), and eliminating cases in which we could not determine grammatical functions adequately, we were left with 2,272 tokens (examples annotated with Supp in the FrameNet corpus), which correspond to 1,093 distinct cases of (verb, noun, frame, FE) tuples.

We provide some examples in Figure 1 for nouns from the Judgment and Judgment_communication frames. Not all LFs are necessarily Oper – some provide semantics of their own. There are several observations we can make.

1. Some nouns have support verbs in the corpus for only one FE (for example *blame*), while other nouns have support verbs for multiple FEs (for example *appreciation* or *scorn*).
2. Some support verbs are used by several nouns: for example, *have* and *make*.
3. Some support verbs are specific to certain nouns, for example *sing praise* but #*sing acclaim*. Of course, it would be conceivable that a larger corpus might include such a collocation, though in this specific instance that seems unlikely.

These observations confirm the fundamental insight into support verbs: they are lexically idiosyncratic, and thus hard to predict. The entire list of extracted support verbs can be found here:

http://www1.ccls.columbia.edu/~nlp/resources/support-verbs.txt

7. Evaluation

We evaluated the algorithm by hand by randomly choosing 208 tuples (types, not tokens – 19% of the data) of keyword nouns, support verbs, the noun's frame, and the verb's subject's FE. We inspected the proposed lexical function and evaluated it, drawing on lexicons and our intuition in order to determine if the relation was in fact an Oper-type lexical function. We classified each pair into one of the following categories:

- Correct. The subject of the verb has the given FE, and the verb contributes no additional element of meaning beyond that contributed by the keyword noun.
  - Example: (*provide*, *assistance*, Assistance, Helper)
  - Example sentence from corpus: We will continue to **provide** substantial financial **assistance** for the protection and preservation of this heritage
- LF with missing semantic component. The subject of the verb has the given FE, but the verb contributes some limited additional element of meaning beyond that contributed by the keyword noun. This additional meaning is typically of the aspectual type: the activity denoted by the keyword noun is just starting, or finishing, or it is caused by the subject of the support verb. MTT provides additional LFs in order to indicate this additional meaning, such as **Incep** for a support verb with the meaning of "beginning", or **CausFunc** for a support verb with the additional meaning of "causation". However, we cannot detect this meaning automatically.
  - Example: (*provoke*, *censure*, Judgment_evaluation, Evaluee)
  - Example: One young lady, disguised as another, would be unlikely therefore to **provoke censure** even if recognized.
  - Analysis: really a CausFunc.
- Data or annotation problem. This is a case where the subject FE is probably mistagged.
- Wrong. These are cases where the tuple does not represent a valid support verb.

The results are as follows:

Good	Correct	158	76%
Semantic Problem	LF with missing semantic component	39	19%
Bad	Data or annotation problem	1	0%
	Wrong	10	5%

This shows that the extraction and semantic classification of collocation material is indeed feasible with semantically annotated corpora. The FrameNet-corpus can well serve as point of departure despite the fact that its Supp annotation is not specifically a major emphasis of the current FrameNet annotation effort.

8. Conclusion

We have shown that we can detect interesting support verb constructions in the current FrameNet annotation. We conclude from this that support verb annotation is feasible, and should be encouraged in semantic annotation projects.

9. Acknowledgements

This paper was written within the framework of the following research project: HUM2005-08052-C02-02 (*Ministerio de Educación y Ciencia* and partially funded by FEDER). We thank also the help provided by the Spanish Ministry (*Ayuda de Movilidad 2007*) which allowed Alonso Ramos and Wanner to stay in Columbia University.

10. References

Benson, M., E. Benson and R. Ilson (1986). *The BBI Combinatory Dictionary of English. A Guide to Word Combinations*. Amsterdam/ Philadelphia: John Benjamins.

Cowie, A. (1993). Phraseology. In: R. E. Asher and J. M. Y. Simpson (Eds.). *The Encyclopedia of Language and Linguistics, Vol. 6*. Oxford: Pergamon Press, pp. 3168–3171.

Crowther, J., S. Dignen and D. Lea (eds.) (2002). *Oxford Collocations Dictionary for Students of English.* Oxford: Oxford University Press.

Fillmore, C.J., Wooters, C. and Baker, C.F. (2001). Building a Large Lexical Databank Which Provides Deep Semantics. In *Proceedings of the Pacific Asian Conference on Language, Information and Computation*. Hong Kong.

Hill, J. and M. Lewis (eds.) (1997). *LTP Dictionary of Selected Collocations*. London: LTP.

Mel'čuk, I. A. (1996). Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In: L. Wanner (Ed.) *Lexical Functions in Lexicography and Natural Language Processing*, Amsterdam/ Philadelphia, PA: Benjamins Academic Publishers, pp. 37–102.

Mel'čuk, I. A., Clas, A. & Polguère, A. (1995). *Introduction à la lexicologie explicative et combinatoire*. Louvain: Duculot.

Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C. R. and Scheffczyk, J. (2006). *FrameNet II: Extended Theory and Practice*. http://framenet.icsi.berkeley.edu/index.php?option=com_wrapper&Itemid=126

Wanner, L. (2004). Towards automatic fine-grained classification of verb-noun collocations. *Natural Language Engineering*. 10(2):95-143.

#	Verb	Noun	FE
1	earn	appreciation	Evaluee
2	express	appreciation	Cognizer
1	obtain	appreciation	Evaluee
1	receive	appreciation	Evaluee
1	show	appreciation	Cognizer
1	voice	appreciation	Cognizer
1	have	approbation	Cognizer
1	attribute	blame	Cognizer
1	heap	blame	Cognizer
2	lay	blame	Cognizer
2	pin	blame	Cognizer
2	place	blame	Cognizer
1	project	blame	Cognizer
4	put	blame	Cognizer
2	have	scorn	Cognizer
1	heap	scorn	Cognizer
4	pour	scorn	Cognizer
1	provoke	scorn	Evaluee
1	have	stricture	Cognizer
1	earn	acclaim	Evaluee
1	receive	acclaim	Evaluee
2	win	acclaim	Evaluee
5	make	accusation	Communicator
1	face	censure	Evaluee
1	incur	censure	Reason
1	provoke	censure	Evaluee
1	draw	condemnation	Evaluee
1	arouse	criticism	Reason
1	direct	criticism	Communicator
1	draw	criticism	Evaluee
1	earn	criticism	Evaluee
1	face	criticism	Reason
1	face	criticism	Communicator
1	face	criticism	Evaluee
1	launch	criticism	Communicator
1	level	criticism	Communicator
2	make	criticism	Communicator
1	prompt	criticism	Reason
1	provoke	criticism	Reason
2	receive	criticism	Communicator
1	receive	criticism	Evaluee
1	target	criticism	Communicator
1	voice	criticism	Communicator
2	attract	praise	Evaluee
1	confer	praise	Communicator
1	draw	praise	Evaluee
1	earn	praise	Evaluee
3	give	praise	Communicator
3	have	praise	Communicator
3	heap	praise	Communicator
2	receive	praise	Evaluee
4	sing	praise	Communicator
2	win	praise	Communicator
2	win	praise	Evaluee
1	employ	ridicule	Communicator
1	pour	ridicule	Communicator

Figure 1: Table of some extracted support verbs. The first column shows the number of examples of this support verb tuple found in the FrameNet corpus. *Appreciation*, *blame*, *scorn* and *structure* are from the Judgment frame, the others from the Judgment_Communication frame.