# KNEWS: Using Logical and Lexical Semantics to Extract Knowledge from Natural Language

**Valerio Basile**[1] and **Elena Cabrio**[2] and **Claudia Schon**[3]

**Abstract.** We present KNEWS, a pipeline of NLP tools that accepts natural language text as input and outputs knowledge in a machine-readable format. The tool outputs frame-based knowledge as RDF triples or XML, including the word-level alignment with the surface form, as well as first-order logical formulae. KNEWS is freely available for download. Moreover, thanks to its versatility, KNEWS has already been employed for a number of different applications for information extraction and automatic reasoning.

## 1 Introduction

*Machine Reading* [7] is the task of extracting formally encoded knowledge from natural language text. It subsumes *Natural Language Understanding*: after resolving the ambiguities in the text, the information extracted is grounded by linking it to a knowledge base. A complete machine reading tool is a step towards the construction of large repositories of general knowledge without having to rely on human-built resources. Such system can automatically learn, for instance, that the `Knife` is used as `Instrument` to `Cut` the `Bread`, where `Knife`, `Instrument`, `Cut` and `Bread` are all entries in some knowledge base on the Web. Moreover, a machine reading component can play an important role in other environments: in many automated reasoning applications, it is necessary to use large ontologies as background knowledge. Machine reading helps to disambiguate predicate names. This supports selection methods like SInE [9] to select very focused background knowledge.

Several tools have been published to solve parts of this complex task, notably the extraction of entities [15, 16], relations [12], semantic roles [10], and more. FRED [14] is a notable example, proposing a complete machine reading pipeline that extracts RDF triples comprising every aspect of the semantics of the input text.

In this demo we present KNEWS, a complete and versatile text-to-knowledge pipeline for machine reading, that tries to overcome some of the limits of existing systems. In particular, in comparison with FRED, from which we drew the initial inspiration, KNEWS is configurable to use different external modules, provides different kind of meaning representations as output, and, last but not least, its source code is freely available and not bound to online APIs. KNEWS is available at `https://github.com/valeriobasile/learningbyreading`, a demo is at `http://gingerbeard.alwaysdata.net/knews/`.

[1] Université Côte d'Azur, Inria, CNRS, I3S, France, {valerio.basile@inria.fr}
[2] Université Côte d'Azur, Inria, CNRS, I3S, France, {elena.cabrio@unice.fr}
[3] Universität Koblenz-Landau, Germany {schon@uni-koblenz.de}

## 2 NLP Pipeline for Knowledge Extraction

KNEWS is a pipeline system. The main components are a semantic parser and two modules for word sense disambiguation and entity linking. KNEWS works by running these components separately on a text, then it aligns the output of the the semantic parser to the output of the other two modules (Figure 1).

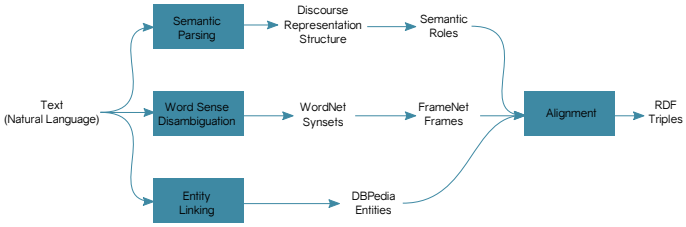


**Figure 1.** Architectural Scheme of KNEWS.

**Semantic parsing** The semantic parsing module must process the input text and output a complete formal representation of its meaning. To this aim, KNEWS employs the *C&C tools* and *Boxer*. The C&C tools [6] are a pipeline of statistical NLP tools including a tokenizer, a lemmatizer, named entity and part-of-speech tagger, and a parser that creates a Combinatorial Caregorial Grammar representation of the natural language syntax. Boxer [5] is a rule-based system that builds an abstract meaning representation on top of the CCG analysis. Such structures contain, among other information, predicates representing the roles of the entities with respect to the detected events, e.g., event(A), entity(B), agent(A,B) to represent B playing the role of the *agent* of the event A.

**Word sense disambiguation and Entity Linking** In order for the semantic representations to be useful in contexts such as knowledge representation and automatic reasoning, the predicates need to be linked to a knowledge base. KNEWS uses WordNet [11] to represent concepts and events, DBpedia [4] for named entities, and FrameNet's frames [1] to represent events, integrating the mapping with the WordNet synsets provided by FrameBase [17]. The inventory of thematic roles used by Boxer is taken from VerbNet [18], while KNEWS employs the mapping provided by SemLinks [13] to link them (whenever possible) to FrameNet roles. By linking the discourse referents representing concepts in a DRS to WordNet synsets, entities to DBpedia and events to FrameNet frames, KNEWS is able to extract complete meaning representations from natural language text linked to Linked Open Data knowledge bases.

[4] `https/dbpedia.org`

## 3 Output Modes

**Frame-based Semantics** The first output mode of KNEWS is frame instances, sets of RDF triples that contain a unique identifier, the type of the frame, the thematic roles involved in the instance, and the concepts or entities that fill the roles. The format follows the scheme of FrameBase [17], which offers the advantage of interoperability with other resources in the Linked Open Data cloud. An example of frame instance, extracted from the sentence "A robot is driving the car." is given by the following triples[5]:

```
fb:fi-dc59afa6 a fb:frame-Operate_vehicle-drive.v .
fb:fi-dc59afa6 fb:fe-Driver dbr:Robot .
fb:fi-dc59afa6 fb:fe-Vehicle wn:02961779-n .
```

This output mode of KNEWS has been employed in [4] to create a repository of general knowledge about objects.

**Word-aligned Semantics** The second output mode of KNEWS is similar to the previous one (frame instances) with the difference that it contains as additional information the alignment with the text. We exploit the Discourse Representation Graph [2] output of Boxer to link the discourse referents to surface forms, i.e., span of the original input text, resulting in a word-aligned representation as:

```
<frameinstances>
  <frameinstance id="9a3fa55e"
      type="Operate_vehicle-drive.v" internalvariable="e1">
    <framelexicalization>k3:x1 is driving k3:x2
    </framelexicalization>
    <instancelexicalization>A robot is driving the car
    </instancelexicalization>
    <frameelements>
      <frameelement role=''Driver'' internalvariable=''x1''>
        <concept>http://dbpedia.org/resource/Robot</concept>
        <roleexicalization>A robot is driving x2
        </roleexicalization>
        <conceptlexicalization/>
      </frameelement>
      <frameelement role=''Vehicle'' internalvariable=''x2''>
        <concept>
          http://wordnet-rdf.princeton.edu/wn31/02961779-n
        </concept>
        <roleexicalization>x1 is driving the car
        </roleexicalization>
        <conceptlexicalization/>
      </frameelement>
    </frameelements>
  </frameinstance>
</frameinstances>
```

Such surface forms can be complete (e.g., "A robot") or incomplete (e.g., "$x_1$ is driving $x_2$") and can be composed to recreate the original, as well as lexicalizations for new knowledge [3].

**First-order Logic** In the third output mode, KNEWS is able to generate first-order logic formulae representing the natural language text given as input. For example the input "She won a spelling bee." leads to the following first-order logic formula:

$$\exists A, B, C(r1Theme(A, C) \wedge r1Agent(A, B) \wedge 201102550\text{-}v(A) \wedge 200940051\text{-}v(C) \wedge r1of(C, B) \wedge 107907011\text{-}n(B) \wedge n2female(B))$$

corresponding to a disambiguated version of the following one:

$$\exists A, B, C(r1Theme(A, C) \wedge r1Agent(A, B) \wedge win(A) \wedge bee(C) \wedge r1of(C, B) \wedge n1spelling(B) \wedge n2female(B))$$

Providing the predicates as Wordnet synsets has the advantage that this information can be exploited to select background knowledge in a much more focused manner, as proposed in [8]. In the example above, the use of KNEWS ensures that "spelling bee" is distinguished from the insect bee. This can be used to avoid selecting background knowledge on insects. The syntax of the formula returned by KNEWS is very similar to the well-known TPTP format [19] — in fact we plan to provide TPTP syntax directly as output of KNEWS.

[5] The instance id and namespaces are abbreviated for readability.

## REFERENCES


[1] Collin F. Baker, Charles J. Fillmore, and John B. Lowe, 'The berkeley framenet project', in *Proceedings of COLING '98*, pp. 86–90, Stroudsburg, PA, USA, (1998). Association for Computational Linguistics.

[2] Valerio Basile, *From logic to language: Natural language generation from logical forms*, Ph.D. dissertation, 2015.

[3] Valerio Basile, 'A repository of frame instance lexicalizations for generation', in *to appear in Proceedings of WebNLG 2016: 2nd International Workshop on Natural Language Generation and the Semantic Web*, (2016).

[4] Valerio Basile, Elena Cabrio, and Fabien Gandon, 'Building a general knowledge base of physical objects for robots', in *The Semantic Web. Latest Advances and New Domains*, (2016).

[5] Johan Bos, 'Wide-coverage semantic analysis with boxer', in *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1, pp. 277–286, (2008).

[6] James R. Curran, Stephen Clark, and Johan Bos, 'Linguistically motivated large-scale nlp with c&c and boxer', in *Proceedings of ACL '07 Poster and Demonstration Sessions*, pp. 33–36, Stroudsburg, PA, USA, (2007). Association for Computational Linguistics.

[7] Oren Etzioni, 'Machine reading at web scale', in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pp. 2–2, New York, NY, USA, (2008). ACM.

[8] Ulrich Furbach and Claudia Schon, 'Commonsense reasoning meets theorem proving', in *to appear in Proceedings of Bridging-20016 - Workshop on Bridging the Gap between Human and Automated Reasoning*, (2016).

[9] Kryštof Hoder and Andrei Voronkov, 'Sine qua non for large theory reasoning', in *Automated Deduction – CADE-23*, eds., Nikolaj Bjørner and Viorica Sofronie-Stokkermans, volume 6803 of *Lecture Notes in Computer Science*, 299–314, Springer Berlin Heidelberg, (2011).

[10] Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson, 'Semantic role labeling: An introduction to the special issue', *Computational Linguistics*, **34**(2), 145–159, (2008).

[11] George A. Miller, 'Wordnet: A lexical database for english', *Commun. ACM*, **38**(11), 39–41, (November 1995).

[12] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky, 'Distant supervision for relation extraction without labeled data', in *Proceedings of ACL09*, pp. 1003–1011, Stroudsburg, PA, USA, (2009). Association for Computational Linguistics.

[13] Martha. Palmer, 'SemLink: Linking PropBank, VerbNet and FrameNet.', in *Proceedings of the Generative Lexicon ConferenceGenLex-09*, Pisa, Italy, (Sept 2009).

[14] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi, 'Knowledge extraction based on discourse representation theory and linguistic frames', in *Proceedings of the EKAW'12*, pp. 114–129, Berlin, Heidelberg, (2012). Springer-Verlag.

[15] Delip Rao, Paul McNamee, and Mark Dredze, *Entity Linking: Finding Extracted Entities in a Knowledge Base*, 93–115, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[16] Lev Ratinov and Dan Roth, 'Design challenges and misconceptions in named entity recognition', in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pp. 147–155, Stroudsburg, PA, USA, (2009). Association for Computational Linguistics.

[17] Jacobo Rouces, Gerard de Melo, and Katja Hose, 'Framebase: Representing n-ary relations using semantic frames.', in *ESWC*, eds., Fabien Gandon, Marta Sabou, Harald Sack, Claudia d'Amato, Philippe Cudr-Mauroux, and Antoine Zimmermann, volume 9088 of *Lecture Notes in Computer Science*, pp. 505–521. Springer, (2015).

[18] Karin Kipper Schuler, *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*, Ph.D. dissertation, Philadelphia, PA, USA, 2005. AAI3179808.

[19] G. Sutcliffe, 'The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0', *Journal of Automated Reasoning*, **43**(4), 337–362, (2009).