On the Automatic Generation of Intermediate Logic Forms for WordNet Glosses

Rodrigo Agerri[1] and Anselmo Peñas[2]

[1] Universidad Politécnica de Madrid (UPM),
Vicomtech Research Centre*
Donostia-San Sebastián, Spain
ragerri@vicomtech.org
[2] NLP & IR Group UNED
Madrid, Spain
anselmo@lsi.uned.es

Abstract. This paper presents an automatically generated Intermediate Logic Form of WordNet's glosses. Our proposed logic form includes neo-Davidsonian reification in a simple and flat syntax close to natural language. We offer a comparison with other semantic representations such as those provided by Hobbs and Extended WordNet. The Intermediate Logic Forms are straightforwardly obtained from the output of a pipeline consisting of a part-of-speech tagger, a dependency parser and our own Intermediate Logic Form generator (all freely available tools). We apply the pipeline to the glosses of WordNet 3.0 to obtain a lexical resource ready to be used as knowledge base or resource for a variety of tasks involving some kind of semantic inference. We present a qualitative evaluation of the resource and discuss its possible application in Natural Language Understanding.

1 Introduction

Ongoing work on text understanding has made clear the need of readily available knowledge and lexical resources that would help systems to perform tasks that involve some type of semantic inference (e.g., [1,2,3]). For example, 21 of 26 teams participating in PASCAL RTE-3 [4] used WordNet as a knowledge resource to support reasoning. It has also been pointed out that we may need to develop deep language understanding techniques if we are to consistently obtain very high performance results in tasks such as RTE [5]. Some work has therefore been done trying to improve the utility of WordNet (notably [6,7]) for semantic inference, by augmenting it with syntactic analysis and logic formalisation of its glosses. This paper reviews previous work aiming to identify those points which could be improved. The result is the development of a new freely available resource consisting of the generation of Intermediate Logic Forms (ILFs) for WordNet 3.0 glosses: ILF-WN.[1]

* Currently at Vicomtech, http://www.vicomtech.org
[1] Freely available to download at http://nlp.uned.es/semantics/ilf/ilf.html

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 26–37, 2010.

The ‘intermediate’ character of ILF comes from the fact that rather than generating a semantic representation in first-order logic (or other type of standard logic), we provide a formal representation that aims to be as close as possible to natural language by performing strict neo-Davidsonian reification [8,9] and reducing to a minimum the syntax complexity. The objective is to provide a flat, syntactically simple formal representation suitable to perform various types of semantic inference (e.g., as in Recognizing Textual Entailment [1]), avoiding the excessive brittleness caused by first-order approaches, as well as being able to tackle difficult semantic problems such as co-reference, anaphora resolution, etc.

Our representation is based on two main predicates, one denoting the existence of a discourse entity *e(Id1,x1)*, and another to identify the existence of a direct relation between two discourse entities *rel(Id1,Id2,x1,x2)*. Both, entities and relations are indexed to easily add semantic information related to the discourse entities (e.g. lexical information: *w(Id1,Word:Pos:Cat), syn(Id1,Synset-offset)*), but also to to the relations (e.g. syntactic dependency types *dep(Id1,Id2,nsubj)*, semantic roles, etc.) in a structure suitable to treat discourse-related problems. For example, co-reference is denoted by the unification of variables in two different discourse entities (e.g. *e(Id1,x1), e(Id3,x1)*).

Next section discusses previous related work. Section 3 describes the main characteristics of Intermediate Logic Forms. Section 4 describes the development of ILF-WN. A qualitative comparison or evaluation with respect to previous approaches to formalise WordNet’s glosses can be found in section 5 and section 6 concludes and points out to any future improvements to ILF-WN.

2 Previous Related Work

Our proposal, both the logic forms and the formalization of WordNet’s glosses is inspired by neo-Davidsonian formalisms used in computational semantics such as [10,11,12]. However, ILF-WN is a flat and simple syntax closer to the output of dependency parsers. The syntax also contemplates that every relation between words is a predicate instead of introducing first-order logical operators. Two approaches have previously offered a logic form of WordNet’s glosses.

2.1 Extended WordNet

Extended WordNet 2.0-1.1 (XWN 2) provides a logical form and sense disambiguation for the glosses of WordNet 2.0 [13,6,12]. A very important feature of XWN is the expansion of WordNet’s relations by taking into account the disambiguated info they extract from the glosses. This is something that current version of ILF-WN does not offer. The overall procedure of building XWN 2 consists of pre-processing the glosses and perform syntactic parsing, logical form transformation and word sense disambiguation (WSD) of adjectives, adverbs, verbs and nouns. They use various methods to perform WSD on the glosses. They disambiguate 64% words of WordNet glosses with 75% accuracy. The rest of the words are tagged with the first sense.

The pre-processing of glosses aims to include the *definiendum* in the *definiens* adding several other terms to make glosses more suitable for syntactic parsing. For example, "the adjective glosses were extended with the adjective and 'is something' in front of the gloss and a period at the end of it" [6]. Take the adjective 'bigheaded': The gloss was transformed from (1) "used colloquially of one who is overly conceited or arrogant; "a snotty little scion of a degenerate family"-Laurent LeSage; "they're snobs–stuck-up and uppity and persnickety", to (2) "bigheaded is something used colloquially of one who is overly conceited or arrogant". The pre-processed gloss is then parsed using Charniak's parser [14] and an in-house parser [6], and its result in a treebank form is included.

The parse results were classified into GOLD (parses manually checked), SILVER (agreement between the two parsers without human intervention) and NORMAL (disagreement between parsers, in-house parser is given priority) qualities (but for formatting reasons, the last element of the tree should be indented to the right of 'VBZ is').

The transformation to logical form [12] is inspired by the eventuality logic proposed by Hobbs [10]. Depending of the part-of-speech of the synset, they use a number of rules for the assignment of variables. In the case of adjectives, "the first word representing the synset is taken and assigned the argument 'x1'. In the gloss of the synset on the right hand size, the argument 'x1' refers to the same entity as the one described by the first word in the synset."

The glosses are included in their original format in English whereas both the parse (Example and logic form (Example 1) elements are performed on the pre-processed versions of the glosses. Furthermore, in the original glosses synsets' definitions, examples and other information is offered as a unit.

Example 1. Logic Form of "bigheaded.s.01" in XWN 2.

```
<lft quality="SILVER">
 bigheaded:JJ(x1) -> use:VB(e1, x6, x1) colloquially:RB(e2)
 of:IN(e1, e2) one:JJ(x3) be:VB(e2, x1) overly:RB(x4)
 conceited:JJ(x4) arrogant:JJ(x4)
</lft>
```

Perhaps due to the glosses pre-processing, some parses result in overly complex structures where in most cases the most important part of the gloss namely, the *definiens*, is buried among a number of subordinating clauses with respect to the phrase 'overly conceited or arrogant'. This problem is fairly frequent for long glosses (usually in nouns), and it seems to degrade the quality of the final logic form. Leaving aside issues such the inclusion of the *definiendum* in the *definiens*, we can see in example 1 that there are variables that do not belong to anything (e.g., *x6*), and others that are left free (not related in any way with the rest of the formula), such as *overly:RB(x4) conceited:JJ(x4) arrogant:JJ(x4)*. Other issues related to the absence of the coordinating disjunction 'or' in the logic form and the assignment of the same variable *x4* for 'overly', 'conceited' and 'arrogant', renders some glosses' logic forms of XWN 2 difficult to understand and use.

2.2 ISI/Boeing WN30lfs

A second project, WN30-lfs, consists of the logical forms for the glosses of WordNet 3.0, except where the parsing failed [7], in XML format, using eventuality notation [10]. It was generated by USC/ISI [7,15]. Every synset is an element consisting of the gloss (without examples, etc.) and its logical form. They preprocessed the glosses to obtain sentences of the form "word is gloss". They parsed them using the Charniak parser [14], and the parse tree is then converted into a logical form by a tool called LFToolkit, developed by Nishit Rathod. In LFToolkit, lexical items are translated into logical clauses involving variables. Finally, as syntactic relations are recognized, variables in the constituents are unified [7]. Furthermore, predicates are assigned word senses using the WordNet semantically annotated gloss corpus [16]. Example 2 shows the logical form for the gloss of *bigheaded*.

Example 2. Logic Form in "bigheaded.s.01" in WN30-lfs.

```
<entry word="bigheaded#a#1" status="partial">
  <gloss>used colloquially of one who is overly conceited or
  arrogant</gloss>
  <lf>bigheaded#a#1'(e0,x0) -> colloquially#r#1'(e5 ,e4) +
     of'(e9,x11,x12) + one'(e11,x12) + excessively#r#1'(e8,e10)
     + conceited#a#1/arrogant#a#1'(e10,x10)</lf>
    <sublf>conceited#a#1'(e10,x10) ->
    conceited#a#1/arrogant#a#1'(e,x10)</sublf>
    <sublf>arrogant#a#1'(e10,x10) ->
    conceited#a#1/arrogant#a#1'(e,x10)</sublf>
</entry>
```

WN30-lfs also includes the sense to be defined in the definition (as in XWN 2) linked by a (seemingly) first-order conditional operator (see Example 2). Furthermore, it is difficult to understand the fact that the logical forms of WN30-lfs often contain free variables and/or predicates without any relation with any other predicates in the definition. As in XWN 2, the predicates for the phrase *overly conceited or arrogant* in Example 2 are left isolated from the rest of the definition.

Summarizing, inspired by XWN 2 and WN30-lfs and acknowledging the many merits of both XWN 2 and WN30-lfs, we believe that there is still some need for providing lexical and/or knowledge resources suitable for computational semantics tasks that required formalized knowledge. In particular, we aim at providing a simple, clear and easy to use logical forms for WordNet's glosses. We also aim at making as transparent as possible the steps taken to obtain the logical forms from the original glosses, and how this information can be offered in a XML structured resource: ILF-WN (Intermediate Logic Form for WordNet glosses).

3 ILF Representation

Our representation consists mainly of two main predicates, one denoting the existence of a discourse entity, and another establishing a direct relation between two discourse entities. Entities and relations are indexed to easily add semantic information related to the discourse entities. In the following subsections we explain this representation in more detail.

3.1 Discourse Entities

Each word introduce a discourse referent denoted by a variable. This variable, together with its index conform the predicate for discourse entities, *e(Id,x)*. The word itself is only a single piece of information associated to the discourse entity among other information obtained during the linguistic processing (e.g. part of speech, lemma, sense, offset in a ontology, similar words, etc). In ILF-WN, we illustrate this with two predicates for lexical information: *w(Id,Word:Pos:Cat)*, *syn(Id,Synset-offset)* (the latter only for monosemous words).

It can be seen that indexes are important to link the lexical information associated to a word with the role of that word in discourse, independently of the variable unification that further reference resolution may produce. In this sense, two discourse entities that denote the same referent will be expressed as *e(Id1,x),e(Id2,x)*. For example, consider the following text from TAC RTE 2009 testset:

> The disappearance of York University chef Claudia Lawrence is now being treated as suspected murder, North Yorkshire Police said. However detectives said **they** had not found any proof that **the 35-year-old**, who went missing on 18 March, was dead. **Her** father Peter Lawrence made a direct appeal to **his** daughter to contact **him** five weeks after **she** disappeared. **His** plea came at a news conference held shortly after a 10,000 reward was offered to help find Miss Lawrence. Crimestoppers said the sum **they** were offering was significantly higher than usual because of public interest in the case.

The pronouns that need to be resolved are highlighted. Using Lingpipe's co-reference system [17], we first identify 'Claudia Lawrence' as a named entity of type PERSON in sentence 1, and then link the female pronoun 'her' in the third sentence to 'Claudia Lawrence' in sentence 1 by assigning them the same reference identifier (*refid*):

```
<coref system="Lingpipe 3.8.1">
<namedent="2" refid="1" s_id="1" type="PERSON" w_ind="6" />
<namedent="4" refid="1" s_id="3" type="FEMALE_PRO" w_ind="1" />
</coref>
```

The co-reference expressions are easily included in the ILFs:

w(1,6,'Claudia_Lawrence','n','nnp') e(1,6,S1_6) w(3,1,'she','prp$','prp$') s(3,1,S1_6)

When the pronoun 'her' (3,1) is resolved to a previous named entity (1,6), then it gets assigned the same variable, namely, word 1 in sentence 3 (her) gets the same discourse referent as word 6 in sentence 1 (Claudia Lawrence). This applies to any subsequent pronouns that are linked to the same entity.

3.2 Relations between Discourse Entities

Strict Davidsonian reification allows us to greatly simplify the syntax of ILF. The relations between entities are introduced by the dependencies obtained by the dependency parser [18]. The predicate *rel(Id1,Id2,x,y)* captures this relation. The pair *Id1,Id2* indexes the relation, preserving the governor-dependent structure of *Id1* with respect to the entity associated to *Id2*. By assigning an index to the relation, we can associate to it any information it might be required (e.g. dependency type, preposition sense, type of noun-noun relation, etc.).

The representation of *buy(x,y)* become *e(Id1,e), rel(Id1,id2,e,x), rel(Id1,id3, e,y)* in our notation. We can then add lexical information, dependency types and semantic roles to the ILF: *w(Id1, Buy), syn(Id1, Syn), dep(Id1,id2,nsubj), dep(Id1,id3,dobj), srl(Id1,id2,Buyer), srl(Id1,id3,Bought).*

In the current version of ILF-WN only the Stanford dependency types are considered, and we include them in the *rel* predicate for simplicity.

4 ILFs for WordNet 3.0 Glosses

4.1 Processing Pipeline

We have assembled a pipeline consisting of a gloss preprocessing module, the C&C tokenizer [19], part-of-speech CRFTagger [20], the Stanford dependency parser [18], and our own ILF generator. ILFs are generated from the dependency parser output adding extra semantic information (if available). The pipeline can take a sentence or discourse in English as an input and automatically generate its ILF. Each third-party tool included in the pipeline is used off-the-self.

Gloss pre-processing. A pre-processing of the glosses was performed in order to obtain grammatically sound sentences more suitable for tokenization, part-of-speech (POS) tagging and syntactic parsing. The pre-processing is loosely inspired in [13]:

1. Text between brackets is removed. Text between brackets is usually an explanation related to the use of the sense defined by the gloss. For example, the gloss of the synset '*bigheaded.s.01*' reads "(used colloquially) overly conceited or arrogant."

2. Everything after a semicolon is removed: Text after the semicolon is usually a semi-structured phrase which does not add anything new to the definition itself. For example, the synset '*concrete.a.01*' is defined as "capable of being perceived by the senses; not abstract or imaginary."
3. According to POS category:
   (a) For nouns and adverbs, we capitalize the first word and add a period at the end. For example, the gloss of the noun '*entity.n.01*' is "That which is perceived or known or inferred to have its own distinct existence."
   (b) For the adjective glosses, 'Something' is added it the beginning and a period at the end of the gloss. The gloss of '*bigheaded.s.01*' mentioned above now reads "Something overly conceited or arrogant." whereas the definition of '*concrete.a.01*' has been transformed to "Something capable of being perceived by the senses."
   (c) The verb glosses were modified by adding 'To' at the beginning of the gloss and a period at the end. The definition of '*swagger.v.03*' is transformed from "act in an arrogant, overly self-assured, or conceited manner" to "To act in an arrogant, overly self-assured, or conceited manner."

Tokenization. The pre-processing performed on the glosses makes it easier for tokenization. We use *tokkie*, the tokenizer offered by the C&C tools [21,19]. Tokenization is performed with *removing quotes* option on.

POS CRFTagger. After tokenization we use the CRFTagger, a Conditional Random Field POS tagger for English [20]. The model was trained on sections 01-24 of Wall Street Journal (WSJ) corpus and using section 00 as the development test set (accuracy of 97.00%) on the Penn Treebank [22]. Even though its reported accuracy is similar to those of Stanford [23] and C&C tools [19] POS taggers also trained on the Penn Treebank, we chose CRFTagger due to its speed in processing large collections of documents.

Dependency parser. We feed the POS tagged glosses to the Stanford Parser [24] in order to output a syntactic analysis consisting of *Stanford typed dependencies*, which amount to a kind of grammatical relations between lemmatized words acting as nodes of a dependency graph [18]. We take advantage of the parser's ability to output the dependency graphs in XML format for a better integration in ILF-WN.

Generation of ILFs. We automatically generate Intermediate Logic Forms from the typed dependencies output of the Stanford Parser, enriching its output with any available lexical and semantic information.

4.2 Description of ILF-WN

Version 0.2 of ILF-WN consists a collection of validated XML documents distributed in two separate packages: (1) Four main files, one per part-of-speech;

(2) a set of files, one per synset, each file identified by its *offset* (a unique identification number for each synset). Both formats contain the part-of-speech (POS), syntactic and ILFs annotations for every gloss in WordNet 3.0.

ILF-WN provides a structured annotation of every gloss in terms of their part-of-speech, syntactic analysis using a dependency parser, and the result of transforming the syntax into an Intermediate Logic Form (ILF). Example 3, which shows the structure of the synset *bigheaded.s.01* in ILF-WN, will be used to describe the resource in more detail.

Being a formalization of WordNet's glosses, ILF-WN is structured in synsets, namely, in senses expressed by a set of synonym words and a gloss. As shown in Example 3 every <sense> element in ILF-WN has three attributes: A unique numeric identifier or *offset*, its POS category in WordNet notation ('a' for *adjectives*, 's' for *satellite adjectives*, 'r' for *adverbs*, 'v' for *verbs* and 'n' for *nouns*), and the synset name, which consists of a lemma, its POS category and the sense number. In Example 3 the synset name is *bigheaded.s.01*, which translates to 'the first sense of the satellite adjective bigheaded'. Decomposing the *offset*, the first digit identifies the POS of the synset, followed by an eight digit number (in the format of the Prolog version of WordNet 3.0 [25]). The first digit of nouns is '1', verb is referred by '2', both adjectives and satellite adjectives are collapsed and start with '3'. Finally, adverbs' offsets start with '4'.

Every <sense> element includes two required and one optional sub-elements: <gloss>, <lemma> (at least one), and <examples> (zero or more). The lemma elements contain the different lemmas of words by which a sense is expressed in WordNet (they are considered synonyms). There might also be some examples of sentences including a use of a word expressing this particular sense. In Example 3, 'bigheaded', 'persnickety', 'snooty', 'snot-nosed', 'snotty', 'stuck-up', 'too_big_for_one's_breeches', and 'uppish', are the 7 lemmas of words that characterize the sense glossed as "Something overly conceited or arrogant". There are also two examples conveying this sense by means of some of the synonym words.

The linguistic annotation specific to ILF-WN is performed on the preprocessed glosses' definitions specified in the <text> element. After tokenizing, POS tagging and dependency parsing, the resulting annotation is placed in the <parse> element in XML format. The dependency graph consists of the POS tagged and lemmatized words of the gloss and the grammatical relations between them. From the dependency graph an ILF is generated and placed in the <ilf> element. For easier readability, we also provide a pretty print of ILF in the <pretty-ilf> element.

5 Comparison with Other Approaches

ILF-WN bears a number of similarities with respect to both XWN 2 and WN30-lfs as its aim, providing lexical knowledge to support semantic inference, fully coincides with their purpose. However, ILF-WN offers a number of particularities added in order to improve the final resource. Although our discussion is based

Example 3. Synset *bigheaded.s.01* in ILF-WN.


```
<sense offset="301890382" pos="s" synset_name="bigheaded.s.01">
  <gloss>
    <text>Something overly conceited or arrogant.</text>
    <parse parser="Stanford parser 1.6.1">
      <s id="1">
        <words pos="true">
          <word ind="1" pos="NN">something</word>
          <word ind="2" pos="RB">overly</word>
          <word ind="3" pos="JJ">conceited</word>
          <word ind="4" pos="CC">or</word>
          <word ind="5" pos="JJ">arrogant</word>
          <word ind="6" pos=".">.</word>
        </words>
        <dependencies style="typed">
          <dep type="advmod">
            <governor idx="3">conceited</governor>
            <dependent idx="2">overly</dependent>
          </dep>
          <dep type="amod">
            <governor idx="1">something</governor>
            <dependent idx="3">conceited</dependent>
          </dep>
          <dep type="amod">
            <governor idx="1">something</governor>
            <dependent idx="5">arrogant</dependent>
          </dep>
          <dep type="conj_or">
            <governor idx="3">conceited</governor>
            <dependent idx="5">arrogant</dependent>
          </dep>
        </dependencies>
      </s>
    </parse>
    <ilf version="0.2">[rel(1,3,2,‘advmod’,G1_3,G1_2),
    rel(1,1,3,‘amod’,G1_1,G1_3), rel(1,1,5,‘amod’,G1_1,G1_5),
    rel(1,3,5,‘conj_or’,G1_3,G1_5), e(1,2,G1_2),
    w(1,2,‘overly’,‘r’,‘rb’), e(1,3,G1_3),
    w(1,3,‘conceited’,‘a’,‘jj’), syn(1,3,301891773), e(1,1,G1_1),
    w(1,1,’something’,‘n’,’nn’), e(1,5,G1_5),
    w(1,5,‘arrogant’,‘a’,‘jj’), syn(1,5,301889819)]</ilf>
    <pretty-ilf>something(x1) amod(x1,x3) amod(x1,x5) overly(x2)
    conceited(x3) advmod(x3,x2) conj_or(x3,x5) arrogant(x5)
    </pretty-ilf>
  </gloss>
  <lemma id="0">bigheaded</lemma>
  <lemma id="1">persnickety</lemma>
  <lemma id="2">snooty</lemma>
  <lemma id="3">snot-nosed</lemma>
  <lemma id="4">snotty</lemma>
  <lemma id="5">stuck-up</lemma>
  <lemma id="6">too_big_for_one’s_breeches</lemma>
  <lemma id="7">uppish</lemma>
  <example id="0">a snotty little scion of a degenerate family-
  Laurent Le Sage</example>
  <example id="1">they’re snobs--stuck-up and uppity and
  persnickety</example>
</sense>
```

on specific examples, most of the points made here are in general applicable to most of the logic forms of WordNet glosses.

First, pre-processing of glosses is a important step to ensure the quality of the resource, specially to remove any redundant and superfluous information from the glosses definitions. Comparing Examples 1 and 2 with 3, it is possible to see that while in Examples 1 and 2 the most relevant concepts (*overly conceited or arrogant*) were somewhat buried among other no so relevant information, in Example 3 it is in a prominent position both in the <parse> and the <ilf> elements.

Second, we have tried to simplify the generation of logical forms with respect to XWN 2 and WN30-lfs, with the objective of avoiding free variables, predicates not related to any other predicates, heterogeneity of the predicates arity, not obvious decisions with respect to the treatment of disjunction, or including the *definiendum* in the *definiens*.

A delicate issue related to logic forms is to decide the argument structure of words, specially verbs with different meanings. In previous representations, this must be specified, requiring some kind of mapping with other resources such as FrameNet [26]. Our representation overcomes this problem by allowing predicates to have its particular argument structure in each particular sentence.

An important feature of XWN 2 and WN30-lfs is the inclusion of word senses in the logical form of glosses. However, in these representations is not possible to consider the complete sense probability distribution of one word, or the different senses coming from different source ontologies. Although we didn't apply any existing disambiguation method to the glosses, the ILF representation proposed here allows to include word sense disambiguation adding the corresponding predicates linked to the corresponding word indexes.

6 Conclusion and Future Work

This paper presents ILF-WN, a freely available XML-structured resource that provides an Intermediate Logic Form for WordNet 3.0 glosses. We have compared ILF-WN with Extended WordNet and WN30-lfs and, while being inspired by them, we aimed to sort out a number of shortcoming presented in those projects. We have also discuss the suitability of ILFs (and of ILF-WN) for the treatment of semantic problems at discourse level.

However, there are several aspects on which ILF-WN has to improve, most notably, on a procedure to include word sense disambiguation [27]. Furthermore, co-reference and anaphora resolution seem to be particularly relevant for noun synsets. For example, the ILF of the (pre-processed) gloss of *blot.n.02*, "An act that brings discredit to the person who does it.", would presumably benefit from resolving the definite description 'the person' to 'who' and 'it' to 'an act'.

ILF-WN could be quantitatively evaluated following the procedure of Task 16 SemEval-2007, for the Evaluation of wide coverage knowledge resources [28]. In this sense it would be similar to the evaluation provided for eXtended Wordnet in [29] where they evaluated XWN's capability of disambiguating words contained in the glosses as reported in section 2.1.

We believe that as we improve ILF-WN towards version 1.0, we will be able to offer both intrinsic (perhaps based on WSD) and extrinsic (based on a task such as RTE [1]) evaluations of the resource.

Acknowledgments

This work has been supported by Madrid R+D Regional Plan, MAVIR Project, S-0505/TIC/000267. (http://www.mavir.net) and by the Spanish Government through the "Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I+D+i 2008-2011 (Grant PR2009-0020).

References

1. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 177–190. Springer, Heidelberg (2006)
2. Clark, P., Murray, W., Thompson, J., Harrison, P., Hobbs, J., Fellbaum, C.: On the role of lexical and world knowledge in RTE3. In: Proceedings of the Workshop on Textual Entailment and Paraphrasing, ACL 2007, Prague, pp. 54–59 (2007)
3. Bos, J., Markert, K.: Recognizing textual entailment with robust logical inference. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 404–426. Springer, Heidelberg (2006)
4. Giampiccolo, D., Magnini, B., Dagan, I., Dollan, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the Workshop on Textual Entailment and Paraphrasing, Association for Computational Linguistics (ACL 2007), Prague, pp. 1–9 (2007)
5. MacCartney, B., Manning, C.: Modeling semantic containment and exclusion in natural language inference. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK, pp. 521–528 (2008)
6. Harabagiu, S.M., Miller, G.A., Moldovan, D.I.: eXtended WordNet - A Morphologically and Semantically Enhanced Resource (2003), `http://xwn.hlt.utdallas.edu`
7. Clark, P., Fellbaum, C., Hobbs, J.R., Harrison, P., Murray, W.R., Thompson, J.: Augmenting WordNet for Deep Understanding of Text. In: Bos, J., Delmonte, R. (eds.) Semantics in Text Processing. STEP 2008 Conference Proceedings. Research in Computational Semantics, vol. 1, pp. 45–57. College Publications (2008)
8. Davidson, D.: Essays on Actions and Events. Oxford University Press, Oxford (1980)
9. Kamp, H., Reyle, U.: From Discourse to Logic: Introduction to Modeltheoretic semantics of natural language, formal language and Discourse Representation Theory. Kluwer Academic Publishers, Dordrecht (1993)
10. Hobbs, J.: Ontological promiscuity. In: Annual Meeting of the ACL, Chicago, pp. 61–69 (1985)
11. Bos, J.: Computational semantics in discourse: Underspecification, resolution, inference. Journal of Logic, Language and Information 13, 139–157 (2004)
12. Rus, V.: Logic Form for WordNet Glosses and Application to Question Answering. PhD thesis, Computer Science Department, School of Engineering, Southern Methodist University, Dallas, Texas (2002)

13. Moldovan, D., Rus, V.: Explaining Answers with Extended WordNet. In: Proceedings of the Association for Computational Linguistics, ACL 2001 (2001)
14. Charniak, E.: A Maximum-Entropy-Inspired Parser. In: Procedings of the North American Association for Computational Linguistics, NAACL (2000)
15. Information Science Institute, University of Southern California: Logical Forms for WordNet 3.0 glosses (2007), http://wordnetcode.princeton.edu/standoff-files/wn30-lfs.zip
16. WordNet Gloss Disambiguation Project, Princeton University: Semantically annotated gloss corpus (2008), http://wordnet.princeton.edu/glosstag.shtml
17. Alias-i: Lingpipe 3.8.2 (2008), http://alias-i.com/lingpipe
18. de Marneffe, M.C., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of Language Resources and Evaluation Conference, LREC (2006)
19. Clark, S., Curran, J.: C&C tools (v1.0), http://svn.ask.it.usyd.edu.au/trac/candc
20. Phan, X.H.: CRFTagger: CRF English POS Tagger (2006), http://sourceforge.net/projects/crftagger
21. Clark, S., Curran, J.: Wide-coverage efficient statistical parsing with CCG and Log-Linear Models. Computational Linguistics 33, 493–553 (2007)
22. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a Large Annotation Corpus of English: The Penn Treebank. Computational Linguistics 19, 313–330 (1993)
23. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of HLT-NAACL, pp. 252–259 (2003)
24. The Stanford Natural Language Processing Group: The Stanford Parser: A statistical parser, http://nlp.stanford.edu/software/lex-parser.shtml
25. Prolog Version of WordNet 3.0 (2008), http://wordnetcode.princeton.edu/3.0/wnprolog-3.0.tar.gz
26. Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C., Sheffczyk, J.: Framenet ii: Extended theory and practice (2006), http://framenet.icsi.berkeley.edu/book/book.html
27. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), Athens, Greece (2009)
28. Cuadros, M., Rigau, G.: Semeval-2007 task 16: Evaluation of wide coverage knowledge resources. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007), Prague, Czech Republic, pp. 81–86. Association for Computational Linguistics (2007)
29. Harabagiu, S., Miller, G., Moldovan, D.: Wordnet 2 - a morphologically and semantic enhanced resource. In: Proceedings of SIGLEX (1999)