Combining Formal and Distributional Models of Temporal and Intensional Semantics

Mike Lewis
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
mike.lewis@ed.ac.uk

Mark Steedman
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
steedman@inf.ed.ac.uk

Abstract

We outline a vision for computational semantics in which formal compositional semantics is combined with a powerful, structured lexical semantics derived from distributional statistics. We consider how existing work (Lewis and Steedman, 2013) could be extended with a much richer lexical semantics using recent techniques for modelling processes (Scaria et al., 2013)—for example, learning that *visiting* events start with *arriving* and end with *leaving*. We show how to closely integrate this information with theories of formal semantics, allowing complex compositional inferences such as *is visiting→has arrived in but will leave*, which requires interpreting both the function and content words. This will allow machine reading systems to understand not just *what* has happened, but *when*.

1 Combined Distributional and Logical Semantics

Distributional semantics aims to induce the meaning of language from unlabelled text. Traditional approaches to distributional semantics have represented semantics in vector spaces (Baroni et al., 2013). Words are assigned vectors based on collocations in large corpora, and then these vectors a composed into vectors representing longer utterances. However, so far there is relatively limited empirical evidence that composed vectors provide useful representations for whole sentences, and it is unclear how to represent logical operators (such as universal quantifiers) in a vector space. While future breakthroughs may overcome these limitations, there are already well developed solutions in the formal semantics literature using logical representations. On the other hand, standard formal semantic approaches such as Bos (2008) have

found that hand-built ontologies such as Word-Net (Miller, 1995) provide an insufficient model of lexical semantics, leading to low recall on applications. The complementary strengths and weaknesses of formal and distributional semantics motivate combining them into a single model.

In Lewis and Steedman (2013), we proposed a solution to these problems which uses CCG (Steedman, 2012) as a model of formal semantics, making it straightforward to build wide-coverage logical forms. Hand built representations are added for a small number of function words such as negatives and quantifiers—but the lexical semantics is represented by first clustering predicates (based on their usage in large corpora), and then using the cluster-identifiers as symbols in the logical form. For example, the induced CCG lexicon might contain entries such as the following[1]:

write $\vdash (S\backslash NP)/NP$
       $: \lambda y \lambda x \lambda e.rel43(x, y, e)$

author $\vdash N/PP_{of}$
       $: \lambda y \lambda x \lambda e.rel43(x, y, e)$

Equivalent sentences like *Shakespeare wrote Macbeth* and *Shakespeare is the author of Macbeth* can then both be mapped to a $rel43(shakespeare, macbeth)$ logical form, using derivations such as:

$$
\frac{
\frac{\text{Shakespeare}}{\frac{NP}{shakespeare}} \quad
\frac{
\frac{\text{wrote}}{\frac{(S\backslash NP)/NP}{\lambda y \lambda x \lambda e.rel43(x,y,e)}} \quad
\frac{\text{Macbeth}}{\frac{NP}{macbeth}}
}{\frac{S\backslash NP}{\lambda x \lambda e.rel43(x, macbeth, e)}}\scriptstyle{>}
}{\frac{S}{\lambda e.rel43(shakespeare, macbeth, e)}}\scriptstyle{<}
$$

This approach interacts seamlessly with standard formal semantics—for example modelling negation by mapping *Francis Bacon didn't write Macbeth* to $\neg rel43(francis\_bacon, macbeth)$. Their method has shown good performance on a dataset of multi-sentence textual inference problems involving quantifiers, by using first-order the-

---

[1] The e variables are Davidsonian event variables.

28

orem proving. Ambiguity is handled by a probabilistic model, based on the types of the nouns.

Beltagy et al. (2013) use an alternative approach with similar goals, in which every word instance expresses a unique semantic primitive, but is connected to the meanings of other word instances using distributionally-derived probabilistic inference rules. This approach risks requiring very large number of inference rules, which may make inference inefficient. Our approach avoid this problem by attempting to fully represent lexical semantics in the lexicon.

2 Proposal

We propose how our previous model could be extended to make more sophisticated inferences. We will demonstrate how many interesting problems in semantics could be solved with a system based on three components:

- A CCG syntactic parse for modelling composition. Using CCG allows us to handle interesting forms of composition, such as co-ordination, extraction, questions, right node raising, etc. CCG also has both a developed theory of operator semantics and a transparent interface to the underlying predicate argument structure.

- A small hand built lexicon for words with complex semantics—such as negatives, quantifiers, modals, and implicative verbs.

- A rich model of lexical semantics derived from distributionally-induced entailment graphs (Berant et al., 2011), extended with subcategories of entailment relations in a similar way to Scaria et al. (2013). We show how such graphs can be converted into a CCG lexicon.

2.1 Directional Inference

A major limitation of our previous model is that it uses a flat clustering to model the meaning of content words. This method enables them to model synonymy relations between words, but not relations where the entailment only holds in one direction—for example, *conquers→invades*, but not vice-versa. This problem can be addressed using the *entailment graph* framework introduced by Berant et al. (2011), which learns globally consistent graphs over predicates in which directed edges indicate entailment relations. Exactly the same methods can be used to build entailment graphs over the predicates derived from a CCG parse:
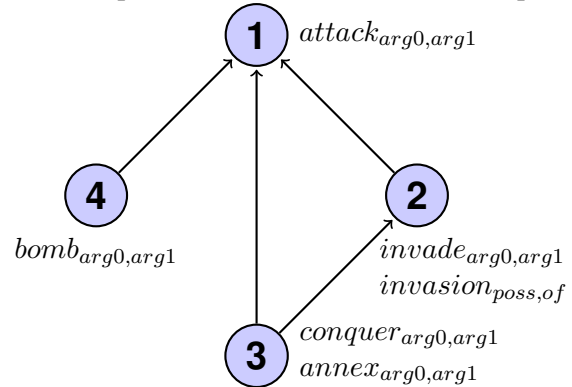
The graph can then be converted to a CCG lexicon by making the semantics of a word be the conjunction of all the relation identifiers it implies in the graph. For example, the above graph is equivalent to the following lexicon:

attack ⊢ (S\NP)/NP
: $\lambda x \lambda y \lambda e.rel1(x, y, e)$

bomb ⊢ (S\NP)/NP
: $\lambda x \lambda y \lambda e.rel1(x, y, e) \wedge rel4(x, y, e)$

invade ⊢ (S\NP)/NP
: $\lambda x \lambda y \lambda e.rel1(x, y, e) \wedge rel2(x, y, e)$

conquer ⊢ (S\NP)/NP
: $\lambda x \lambda y \lambda e.rel1(x, y, e) \wedge$
$rel2(x, y, e) \wedge rel3(x, y, e)$

This lexicon supports the correct inferences, such as *conquers→attacks* and *didn't invade→didn't conquer*.

2.2 Temporal Semantics

One case where combining formal and distributional semantics may be particularly helpful is in giving a detailed model of temporal semantics. A rich understanding of time would allow us to understand *when* events took place, or when states were true. Most existing work ignores tense, and would treat the expressions *used to be president* and *is president* either as equivalent or completely unrelated. Failing to model tense would lead to incorrect inferences when answering questions such as *Who is the president of the USA?*

Another motivation for considering a detailed model of temporal semantics is that understanding the time of events should improve the quality of the distributional clustering. It has recently been shown that such information is extremely useful for learning equivalences between predicates, by determining which sentences describe the same

events using date-stamped text and simple tense heuristics (Zhang and Weld, 2013). Such methods escape common problems with traditional approaches to distributional similarity, such as conflating causes with effects, and may prove very useful for building entailment graphs.

Temporal information is conveyed by both by auxiliary verbs such as *will* or *used to*, and in the semantics of content words. For example, the statement *John is visiting Baltimore* licences entailments such as *John has arrived in Baltimore* and *John will leave Baltimore*, which can only be understood through both knowledge of tense and lexical semantic relations.

The requisite information about lexical semantics could be represented by labelling edges in the entailment graphs, along the lines of Scaria et al. (2013). Instead of edges simply representing entailment, they should represent different kinds of lexical relations, such as *precondition* or *consequence*. Building such graphs requires training classifiers that predict fine-grained semantic relations between predicates, and defining transitivity properties of the relations (e.g. a precondition of a precondition is a precondition). For example, the system might learn the following graph:
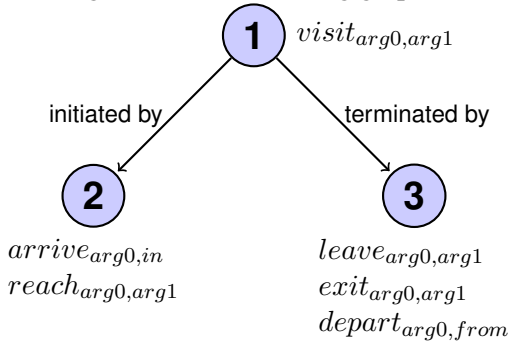
By defining a simple mapping between edge labels and logical forms, this graph can be converted to CCG lexical entries such as:

visit $\vdash (S\backslash NP)/NP$
$\quad : \lambda y\lambda x\lambda e.rel1(x,y,e) \wedge$
$\quad \exists e'[rel2(x,y,e') \wedge before(e,e')] \wedge$
$\quad \exists e''[rel3(x,y,e'') \wedge after(e,e'')]$

arrive $\vdash (S\backslash NP)/PP_{in}$
$\quad : \lambda y\lambda x\lambda e.rel2(x,y,e)$

leave $\vdash (S\backslash NP)/NP$
$\quad : \lambda y\lambda x\lambda e.rel3(x,y,e)$

These lexical entries could be complemented with hand-built interpretations for a small set of common auxiliary verbs:

has $\vdash (S\backslash NP)/(S_b\backslash NP)$
$\quad : \lambda p\lambda x\lambda e.before(r,e) \wedge p(x,e)$

will $\vdash (S\backslash NP)/(S_b\backslash NP)$
$\quad : \lambda p\lambda x\lambda e.after(r,e) \wedge p(x,e)$

is $\vdash (S\backslash NP)/(S_{ng}\backslash NP)$
$\quad : \lambda p\lambda x\lambda e.during(r,e) \wedge p(x,e)$

used $\vdash (S\backslash NP)/(S_{to}\backslash NP)$
$\quad : \lambda p\lambda x\lambda e.before(r,e) \wedge p(x,e) \wedge$
$\quad \neg\exists e'[during(r) \wedge p(x,e')]$

Here, r is the reference time (e.g. the time that the news article was written). It is easy to verify that such a lexicon supports inferences such as *is visiting→will leave*, *has visited→has arrived in*, or *used to be president→is not president*.

The model described here only discusses tense, not aspect—so does not distinguish between *John arrived in Baltimore* and *John has arrived in Baltimore* (the latter says that the consequences of his arrival still hold—i.e. that he is still in Baltimore). Going further, we could implement the much more detailed proposal of Moens and Steedman (1988). Building this model would require distinguishing *states* from *events*—for example, the semantics of *arrive*, *visit* and *leave* could all be expressed in terms of the times that an *is in* state holds.

2.3 Intensional Semantics

Similar work could be done by subcategorizing edges in the graph with other lexical relations. For example, we could extend the graph with *goal* relations between words, such as between *set out for* and *arrive in*, *search* and *find*, or *invade* and *conquer*:
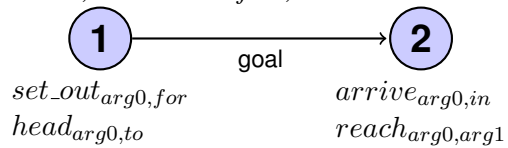
The corresponding lexicon contains entries such as:

set out $\vdash (S\backslash NP)/PP_{for}$
$\quad : \lambda y\lambda x\lambda e.rel1(x,y,e) \wedge$
$\quad \diamond\exists e'[goal(e,e') \wedge rel2(x,y,e')]$

The modal logic $\diamond$ operator is used to mark that the goal event is a hypothetical proposition, that is not asserted to be true in the real world—so *Columbus set out for India*$\not\rightarrow$*Columbus reached India*. The same mechanism allows us to handle Montague (1973)'s example that *John seeks a unicorn* does not imply the existence of a unicorn.

Just as temporal information can be expressed by auxiliary verbs, relations such as goals can

$$
\begin{array}{cccc}
\text{Columbus} & \text{failed} & \text{to} & \text{reach India} \\
\hline
S_{dcl}/(S_{dcl}\backslash NP) & (S_{dcl}\backslash NP)/(S_{to}\backslash NP) & (S_{to}\backslash NP)/(S_b\backslash NP) & S_b\backslash NP \\
\lambda p.p(Columbus) & \lambda p\lambda x\lambda e.\diamond\exists e'[p(x,e')\wedge goal(e',e)]\wedge\neg\exists e''[p(x,e'')] & \lambda p\lambda x\lambda e.p(x,e) & \lambda x\lambda e.rel2(x,India,e)
\end{array}
$$

Figure 1: Output from our system for the sentence *Columbus failed to reach India*

be expressed using implicative verbs like *try* or *fail*. As the semantics of implicative verbs is often complex (Karttunen, 1971), we propose hand-coding their lexical entries:

try $\vdash (S\backslash NP)/(S_{to}\backslash NP)$
  $: \lambda p\lambda x\lambda e.\diamond\exists e'[goal(e,e')\wedge p(x,e')]$

fail $\vdash (S\backslash NP)/(S_{to}\backslash NP)$
  $: \lambda p\lambda x\lambda e.\diamond\exists e'[goal(e,e')\wedge p(x,e')]\wedge$
  $\neg\exists e''[goal(e,e'')\wedge p(x,e'')]$

The $\diamond$ operator is used to assert that the complement of *try* is a hypothetical proposition (so *try to reach*$\not\rightarrow$*reach*). Our semantics for *fail* is the same as that for *try*, except that it asserts that the goal event did not occur in the real world.

These lexical entries allow us to make complex compositional inferences, for example *Columbus failed to reach India* now entails *Columbus set out for India*, *Columbus tried to reach India* and *Columbus didn't arrive in India*.

Again, we expect that the improved model of formal semantics should increase the quality of the entailment graphs, by allowing us to only cluster predicates based on their real-world arguments (ignoring hypothetical events).

3 Conclusion

We have argued that several promising recent threads of research in semantics can be combined into a single model. The model we have described would enable wide-coverage mapping of open-domain text onto rich logical forms that model complex aspects of semantics such as negation, quantification, modality and tense—whilst also using a robust distributional model of lexical semantics that captures the structure of events. Considering these interwined issues would allow complex compositional inferences which are far beyond the current state of the art, and would give a more powerful model for natural language understanding.

References

M. Baroni, R. Bernardi, and R. Zamparelli. 2013. Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technologies*.

Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. 2013. Montague meets markov: Deep semantics with probabilistic logical form. pages 11–21, June.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 610–619. Association for Computational Linguistics.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Research in Computational Semantics, pages 277–286. College Publications.

L. Karttunen. 1971. *The Logic of English Predicate Complement Constructions*. Linguistics Club Bloomington, Ind: IU Linguistics Club. Indiana University Linguistics Club.

Mike Lewis and Mark Steedman. 2013. Combined Distributional and Logical Semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

G.A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational linguistics*, 14(2):15–28.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In *Approaches to natural language*, pages 221–242. Springer.

Aju Thalappillil Scaria, Jonathan Berant, Mengqiu Wang, Peter Clark, Justin Lewis, Brittany Harding, and Christopher D. Manning. 2013. Learning biological processes with global constraints. In *Proceedings of EMNLP*.

Mark Steedman. 2012. *Taking Scope: The Natural Semantics of Quantifiers*. MIT Press.

Congle Zhang and Daniel S Weld. 2013. Harvesting parallel news streams to generate paraphrases of event relations.