Web Question Answering by Exploiting Wide-Coverage Lexical Resources

Michael Kaisser
M.Kaisser@sms.ed.ac.uk

This poster describes an approach to Question Answering that uses the linguistic information available in lexical resources like FrameNet, PropBank or VerbNet to find on the web, answers to natural language questions. The approach is realized in a system I am currently developing.

Motivation

Lexical resources like FrameNet, PropBank or VerbNet attempt to document the range of semantic and syntactic combinatory possibilities (valences) of each word or its senses. They can thus be used to analyse natural language sentences or texts and to gain a deeper understanding of those.

This work aims to exploit these data collections in order to find answers to user questions on the web.

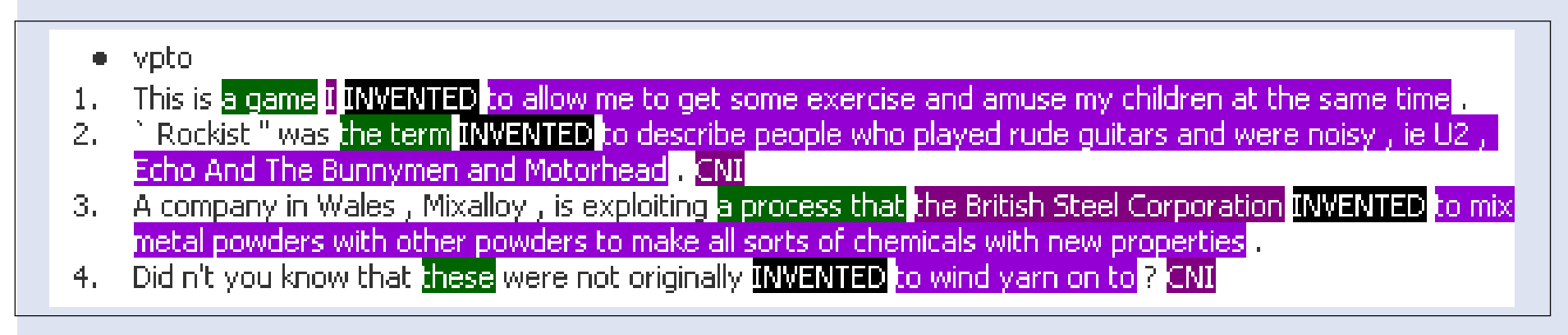
The use of the aforementioned resources can help to overcome a major problem of natural language data collections (such as the Web), it's variability: One and the same fact can be expressed in a lot of different ways.

Lexical Resources

FrameNet

FrameNet is a lexical database resource based on frame semantics and supported by corpus evidence. It contains human-annotated sentences (currently more than 135,000), which exemplify the use of currently 7,388 lexical units organized into 625 semantic frames.

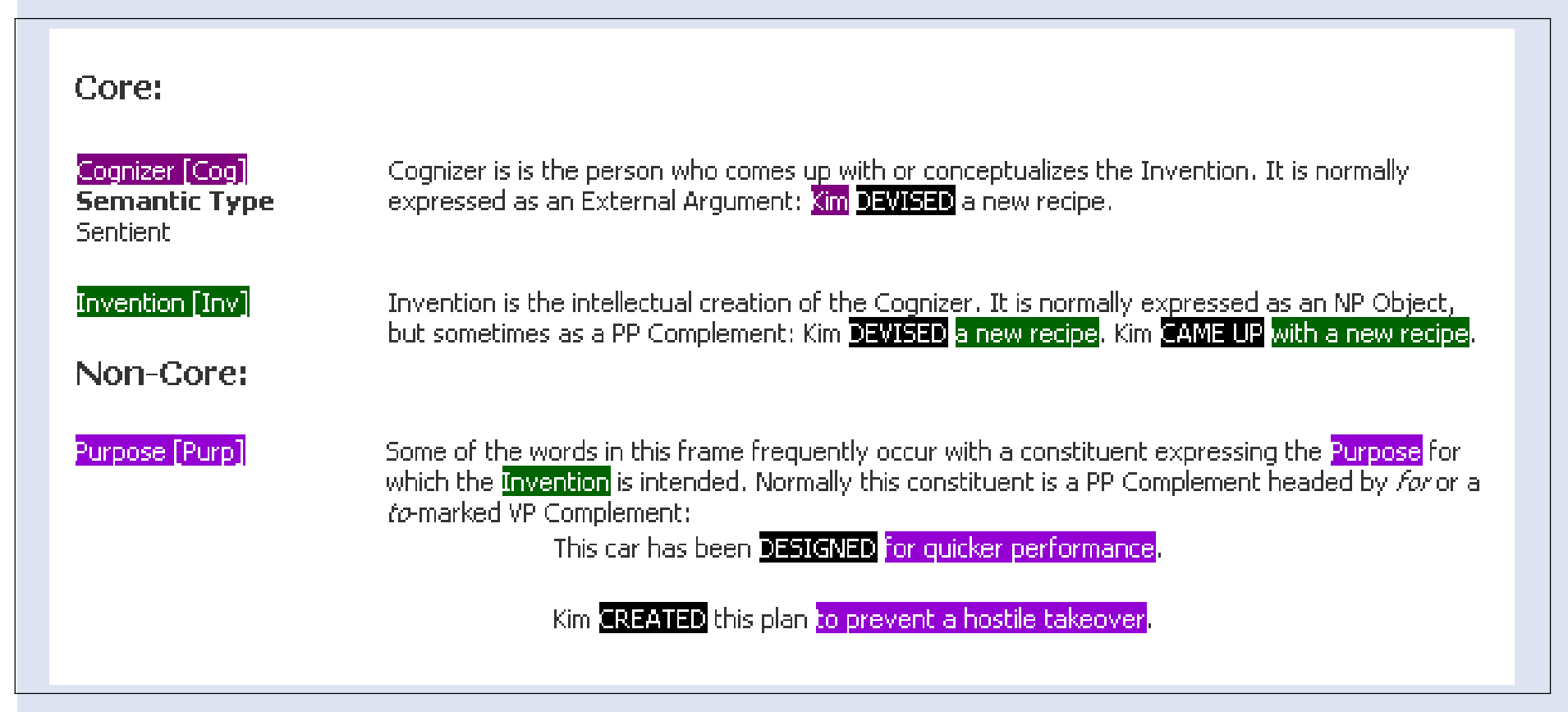
Example of annotated sentences in FrameNet:



- vpto

1. This is a game I INVENTED to allow me to get some exercise and amuse my children at the same time .
2. ` Rockist " was the term INVENTED to describe people who played rude guitars and were noisy , ie U2 , Echo And The Bunnymen and Motorhead . CNI
3. A company in Wales , Mixalloy , is exploiting a process that the British Steel Corporation INVENTED to mix metal powders with other powders to make all sorts of chemicals with new properties .
4. Did n't you know that these were not originally INVENTED to wind yarn on to ? CNI

Legend:



Core:

Cognizer [Cog] **Semantic Type** Sentient	Cognizer is is the person who comes up with or conceptualizes the Invention. It is normally expressed as an External Argument: Kim DEVISED a new recipe.
Invention [Inv]	Invention is the intellectual creation of the Cognizer. It is normally expressed as an NP Object, but sometimes as a PP Complement: Kim DEVISED a new recipe. Kim CAME UP with a new recipe.

Non-Core:

Purpose [Purp]	Some of the words in this frame frequently occur with a constituent expressing the Purpose for which the Invention is intended. Normally this constituent is a PP Complement headed by *for* or a *to*-marked VP Complement: This car has been DESIGNED for quicker performance. Kim CREATED this plan to prevent a hostile takeover.

Lexical units in the *Invention* frame:

coin.v, come up.v, conceive.v, concoct.v, concoction.n, contrivance.n, contrive.v, cook_up.v, create.v, design.n, design.v, ...

For Comparison: VerbNet

VerbNet does not list annotated example sentences, but basic syntactic frames for every entry.

Example frames for *create-26.4*:

- NP[Agent] V NP[Product]
- NP[Agent] V NP[Product] for NP[Beneficiary]
- NP[Agent] V NP[Product] from NP[Material]
- NP[Agent] V NP[Product] as NP[Predicate]

Other members of this VerbNet class:

coin, compose, compute, concoct, construct, contrive, create, derive, fabricate, form, ...

"Who invented the telegraph?" – An Example

First, the incoming question is parsed using MiniPar, and the resulting dependency tree is simplified to the following structure:

```
head      : invented(V)
head\subj : Who
head\obj  : the telegraph
```

↓

This provides enough information to look up the head verb in–for example–FrameNet, where two lexical units for *invent.v* can be found, containing annotated sentences like the following:

Du Pont FE:Cognizer	in the USA	had	INVENTED lexical unit

nylon FE:Invention	in the late 1930s

↓

This is then simplified to a abstract, VerbNet-like frame (by omitting certain constituents):

```
NP[Cognizer,Subj] VERB NP[Invention,Obj]
```

↓

By taking a look at the grammatical functions, it can be concluded that the filler for the *Invention* frame element must be "the telegraph", and that the question asks for a *Cognizer*. The system can now construct a pseudo-semantic formula for the question:

```
invent_272(Cognizer  = X,
           Invention = "the telegraph")
```

↓

Furthermore, the abstract frame can be put in any desired tense:

```
NP[X] invented NP[the telegraph]
NP[X] has invented NP[the telegraph]
NP[X] had invented NP[the telegraph]
```

↓

From this the system generates search engine queries, e.g.:

```
"had invented the telegraph"
```

↓

The search engine will return snippets containing sentences which are parsed, e.g.:

```
"By 1832 Samuel FB Morse
  had invented the telegraph."
```

↓

Because the system knows where in that sentence the answer is located, it can now be extracted. For the above sentence it must be the NP preceding "had", thus:

```
"Samuel FB Morse"
```

↓

For the given example, the system was able to find the correct, exact answer and the open proposition from above can now be completed:

```
invent_272(Cognizer = "Samuel FB Morse",
           Invention = "the telegraph")
```

Further Work

Answer Types

An important component of QA systems is concerned with checking that answers are of the correct semantic type: QA systems usually know a question like "When was Franz Kafka born?" should be answered with a date, while "Who invented the telegraph?" asks for a person.

Fillers for FE *Cognizer*

Count	Name
Pronouns:	
861	Pronoun sing.
266	Pronoun pl.
Named Entities:	
198	Person
22	Organization
16	Location
WordNet:	
2205	entity (id: 1740)
1232	object (id: 16236)
794	living thing (id: 3009)
794	organism (id: 3226)
776	causal agent (id: 5598)
758	person (id: 6026)
459	group (id: 26769)
323	social group (id: 7470450)
217	organization (id: 7523126)
203	artifact (id: 19244)

Fillers for FE *Invention*

Count	Name
Pronouns:	
0	Pronoun sing.
4	Pronoun pl.
Named Entities:	
0	Person
0	Organization
0	Location
WordNet:	
287	entity (id: 1740)
226	object (id: 16236)
162	abstraction (id: 16236)
110	relation (id: 27929)
107	psychol. feature (id: 20333)
103	cognition (id: 20729)
98	whole (id: 2645)
98	artifact (id: 19244)
97	social relation (id: 28549)
97	communication (id: 28764)

It is planned to develop a type checking approach based on an analysis of frame element fillers. The basic assumption is that most frame elements have dedicated semantic classes that their fillers can come from. The above figure shows the results of an experiment done to test this assumption (based on the FrameNet data).

Role Assignment

The correct interpretation of the question–i.e. the detection of the lexical unit to look up and the correct assignment of parts of the question to frame elements–is crucial for the sketched approach. In recent years, a lot of work has been done on automatic labeling of semantic roles. However, a notable difference in this approach is, that here it is solely necessary to annotated questions with semantic roles.

This difference is crucial for at least two reasons:

- Questions tend to be shorter and show a smaller range of syntactic variants than declarative sentences.
- There always is one semantic role that has to be annotated which is not mentioned in the question, but which is very important, because it represents the answer to the question.

I hope to be able to develop accurate, non-statistical methods, based on a syntactic analysis of the question.

Word-Sense Disambiguation

Word-sense disambiguation in this context mainly means deciding between different lexical units that might exist for the word in the question that needs to be looked up.

Again, a lot of research on WSD has been done so far, but here the problem needs to be adjusted to questions, where only a few words are available which can serve as hints to pick the correct sense.