Proper and Efficient Treatment of Anaphora and Long-Distance Dependency: an Experiement with Medical Text

Wai Lok TAM[1], Yusuke MATSUBARA[1], Koiti HASIDA[1], Motoyuki TAKAAI[2], Eiji ARAMAKI[3], Mai MIYABE[3], and Hiroshi UOZAKI[4]

[1]Socia ICT Center, Graduate School of Information Science and Technology, The University of Tokyo
[2]Communication Technology Laboratory, Research and Technology Group, Fuji Xerox Co., Ltd.
[3]Design School, University of Kyoto
[4]Department of Pathology, School of Medicine, Teikyo University

1 Introduction

This paper is a follow-up to (Hasida et al.2012)'s work on pathological reports, a kind of medical text. Such reports have the following characteristics:

1. The composition of such reports has to follow a published set of strict guidelines (In our case, these guidelines are given in (JGCA2010). English version of these guidelines can be found in (JGCA2011).)

2. The subject matters of such reports are strictly limited to specimens submitted for pathological analysis.

These characteristics put the text in pathological reports under the category of controlled natural language, making it a better object text for semantic analysis and knowledge representation. Readers unfamiliar with controlled natural language are recommended to check the survey by (Schwitter2010).

The purpose of this paper is to present how to combine a CFG (Context-Free Grammar) with an ontology to account for both syntactic structures and semantic structures of sentences (and discourses) containing long-distance dependencies and anaphora found in pathological reports. The syntactic and semantic framework outlined in this paper are developed on the foundation of the Global Document Annotation (GDA) guidelines proposed by (Hasida2010).

When constructing our grammar, we have an application in mind. This application is auto-completion and hence speed matters. We want to do a bit more than bigrams can achieve with auto-completion such that the effect of an antecedent or a relative clause on user input can be captured. It is true that an elaborated feature structure-based grammars with hundreds of features would have little problem with anaphora and long distance dependencies. But speed is a problem for such a grammar. This leaves us with CFGs but typical CFGs can handle neither of the phenomena we are interested in. So we make CFGs do the job.

2 Components of Our Grammar

Essentially, our grammar works like a simple unification-based grammar. For illustrative purpose, let us first explain how it works as if it is a unification based grammar represented by directed acyclic graphs (DAGs). The nodes in one such graph are either concepts taken from an ontology or relations between them.

A baby version of our ontology trimmed down to a hierarchy of concepts relevant to examples sentences cited in this paper is given in figure 1.

International Joint Conference on Natural Language Processing Workshop on Natural Language Processing for Medical and Healthcare Fields, pages 33–37, Nagoya, Japan, 14-18 October 2013.

(1) [$_N$ 小弯長] [$_{NOADJ\text{-}GA}$ 12cm] [、,]
syouwantyou zyuni_senti COMMA
[$_N$ 大弯長] [$_{NOADJ\text{-}GA}$ 19.5cm] [。。]
daiwantyou zyukyutengo_senti PERIOD

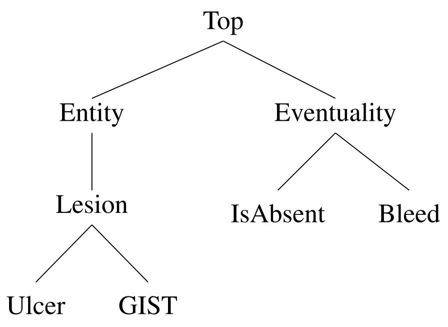

Figure 1: Hierarchy of Concepts Used in Example Sentences

Next comes the links between concepts in our ontology and words in our example sentences. These links are given in the lexicon illustrated by figure 2.

The links between concepts in our ontology and the meaning of a sentence are computed by a handful of semantic composition rules, the most fundamental of which is the rule for headed structure given in figure 3.

In figure 3, the mother(M), the head daughter (HD) and the nonhead daughter (ND) are determined by the combination of POS labels in the syntactic rule given below:

$$S \rightarrow N\phi \ \underline{\text{S-GA}}$$

The underlined daughter is the head daughter of the mother on the left hand side.

3 How the Components Work Together to Parse a Simple Sentence

Now let us parse an example sentence (1) with the syntactic rules. The parse tree is given in figure 4.

To make sense of the semantic composition going on here, some explanation for the path labels and the node

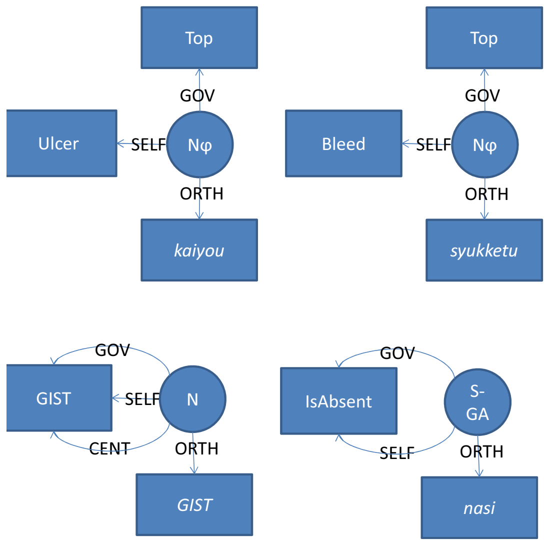

Figure 2: Lexicon

labels is probably needed. The path labels in upper cases SELF and GOV are fundamental to all constituents. The two ends of a SELF path are the P(art) O(f) S(peech) of a constituent and the meaning of the constituent. The two ends of a GOV path are the POS of a constituent and the meaning of the head on which the constituent depends. If a GOV path connects the same nodes as a SELF path, this means the constituent in question is not a dependent of any other constituent. The path label "theme" is a relation from the Top concept to the Abnormality concept defined in our ontology.

When the node labelled "*Top*" and connected by the GOV path to the Nϕ "*kaiyou*" unifies with the node labelled "*IsAbsent*" and connected to SELF path to the S-GA "*nasi*" as a result of the rule illustrated in figure 3, the "*IsAbsent*" concept is also specified as the domain of

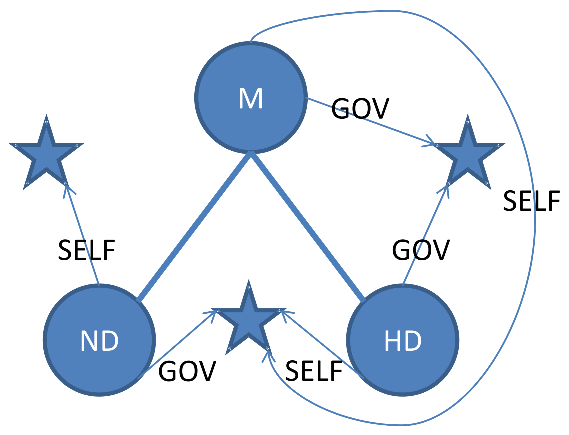

Figure 3: Semantic Rule for Headed Structures

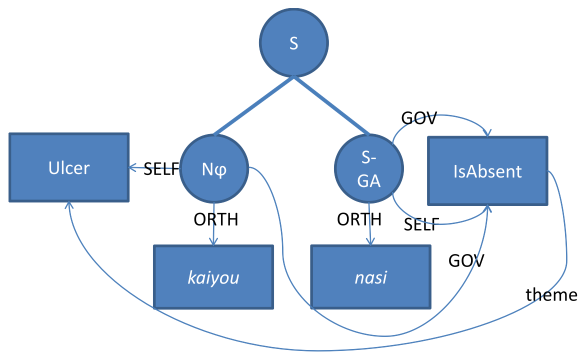

Figure 4: Parse Tree of an Example of Headed Structure

the "*theme*" relation. This way we connect the "*IsAbsent*" concept to the "*Ulcer*" concept by the "*theme*" relation, yielding the semantic representation of the example sentence "*Kaiyou Nasi*", which means "No ulcer is found".

4 Dealing with Anaphora

Now let us substitute the Nϕ in our example sentence with the verbal noun "*syukketu*", meaning "bleed". This introduces a gap in the sentence and the gap refers to the subject of "*syukketu*", which is nowhere to be found in the sentence. To resolve the zero anaphora, we need to store this gap somewhere. Meeting this need is one of the purposes of the CENT path we would like to introduce here. The CENT path also serves the need to pass up the value of the node it connects the POS node to such that both the antecedent and the anaphora can see each other. The percolation is handled by the rule illustrated in figure 5. Applying this rule to our example sentence yields the parse tree given in figure 6.

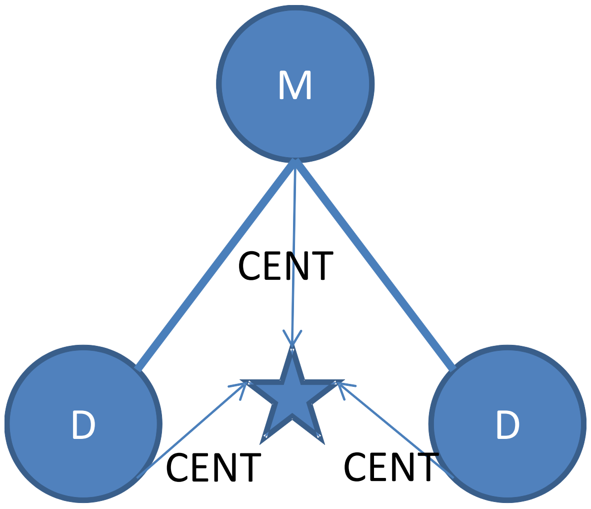

Figure 5: Percolation of Zero Anaphora

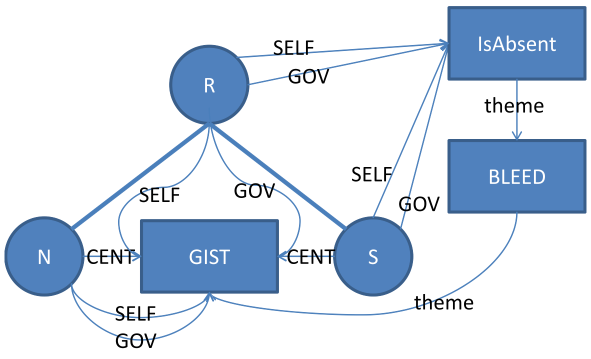

Figure 6: Parse Tree of an Example Containing an Anaphoric Expression

After passing the anaphoric gap to the root, we now come to the point to resolve the anaphora. In order to do this, we need an anaphora resolution rule, which is illustrated in figure 7 and another sentence containing the antecedent. Let us keep this sentence simple and make it constitute of a N(oun) "GIST", meaning "Gastrointestinal Stromal Tumor". When acting as an antecedent, the node connected by the CENT path to the POS node of it shares the same value with the node connected by the SELF path to the POS node, as illustrated in figure 2. We also need to add a syntactic rule for combining a N with a S to form a R(eport).

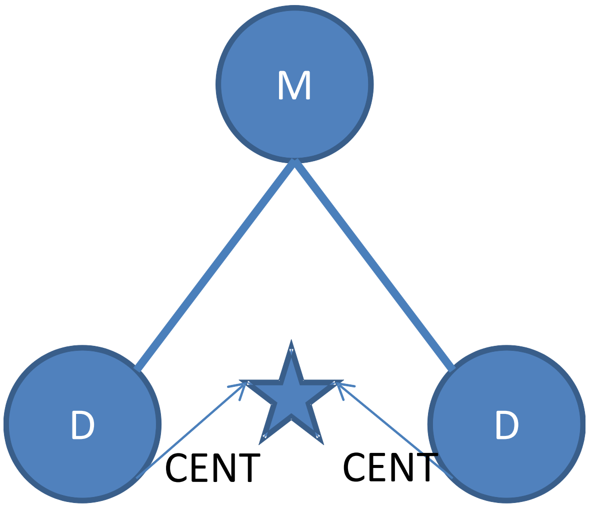

Figure 7: Resolving an Anaphora

$$\text{R} \rightarrow \underline{\text{N}}\ \underline{\text{S}}$$

The semantic composition that goes hands in hand with this syntactic rule is illustrated in figure 8. Some parts of the semantic composition have nothing to do with anaphora resolution. They are there to make sure that the meaning of the mother R is the sum of the meaning of its parts. This is done by introducing two nodes connected by the SELF path to the R node and two nodes connected by the GOV node to the R node. So we have a pair of nodes for each daughter to pass up the values assigned to the pairs of nodes connected to it by its SELF path and GOV path. Putting everything together, we get the parse tree illustrated in figure 9.

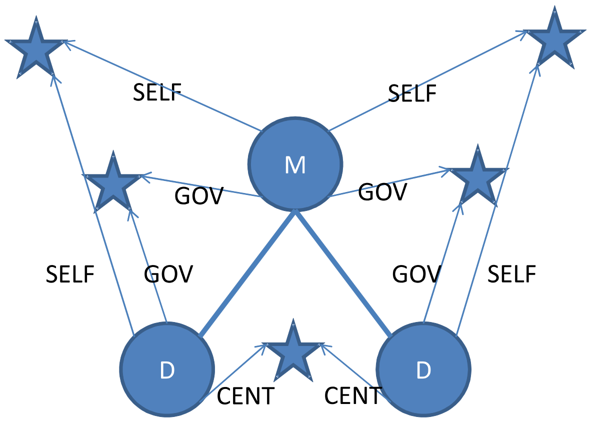

Figure 8: Conjoining Two Sentences and Resolving an Anaphora

5 Compiling DAGs into CFG rules

Now let us turn what we present so far into a CFG. The first step is to merge all nodes connected to the POS node by various paths into a single label in a predefined order such that the order can tell which value corresponds to which node. Assuming that the values are ordered: POS|CENT|SELF|GOV, the lexical entries illustrated in figure 2 are rewritten as:

$$\begin{aligned}
\text{N}\phi|\_|Ulcer|Top &\rightarrow "kaiyou" \\
\text{N}\phi|\_|Bleed|Top &\rightarrow "syukketu" \\
\text{N}|GIST|GIST|GIST &\rightarrow "GIST" \\
\text{S-GA}|\_|IsAbsent|IsAbsent &\rightarrow "nasi"
\end{aligned}$$

The two syntactic rules, which are typical CFG rules, would have to be rewritten as follows such that CFG rules can do the magic of semantic composition:

$$\begin{aligned}
\text{S}|\_|IA|IA &\rightarrow \text{N}\phi|\_|U|T\ \ \underline{\text{S-GA}}|\_|IA|IA \\
\text{S}|T|IA|IA &\rightarrow \text{N}\phi|T|B|T\ \ \underline{\text{S-GA}}|\_|IA|IA \\
\text{R}|\_|G,IA|G,IA &\rightarrow \text{N}|G|G|G\ \ \text{S}|T|IA|IA
\end{aligned}$$

where "IA" stands for "*IsAbsent*", "U" stands for "*Ulcer*", "B" stands for "*Bleed*", "T" stands for "*Top*" and "G" stands for "*GIST*".

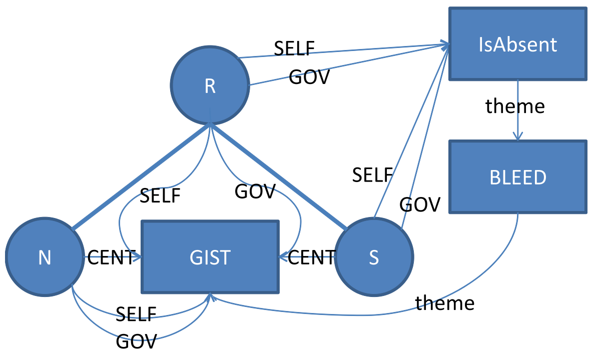

Figure 9: Parse Tree of a Report Containing an Anaphora and the Antecedent it refers to

When the lexicon contains only words referring to leaf concepts, this set generates the same set of sentences as the unification based grammar presented earlier on. If we add the N ”*byouhen*”, which is assigned the superclass of GIST and Ulcer, Lesion as the meaning of it to the lexicon, the CFG given here becomes no longer equivalent to the unification based grammar presented in the beginning of this section. So we need to take the step of expanding the rules to cover words denoting ancestors or descendants of the CENT values, the SELF values and GOV values that make up parts of symbols in a CFG rule. This step adds the following rules to our CFG:

$$\begin{aligned} \mathrm{S}|\_|IA|IA &\rightarrow \mathrm{N}\phi|\_|L|T \quad \underline{\text{S-GA}}|\_|IA|IA \\ \mathrm{R}|\_|L,IA|L,IA &\rightarrow \mathrm{N}|L|L|L \quad \mathrm{S}|T|IA|IA \end{aligned}$$

where ”L” stands for ”*Lesion*”.

6 Conclusion and Future Work

The point of giving the details of compiling a unification based grammar into a CFG is to make the beauty of the small number of features to stand out. Assuming a rule with m features, each having n possible values, adding k values to an existing feature would increase the number of rules by $k \cdot (m-1) \cdot n$. Assigning k values to a new feature, would increase the number of rules by $k \cdot m \cdot n$. So a strictly limited number of features speed up things. This adds to the speed resulting from the lower complexity value $O(n^3)$ of a CFG when compared to the exponential complexity of a feature structure based grammar. The combined power of our grammar design and compilation into CFG make it possible for us to answer the needs of a real time task like auto-completion. Without any optimization, we achieve $500ms$ per sentence when running our grammar on a parser written in Python, an interpreted language. This is pretty much the best a deep parser can achieve as reported by (Matsuzaki et al.2007) on an older but likely to be faster machine than ours, a notebook computer with a 2.40Ghz Core 2 Duo processor. We are hopeful that we can further improve our speed by simply implementing our parser in a compiled language. When it is done, our grammar is expected to support auto-completion of less controlled natural languages such as the language used in nursing reports.

References

Koiti Hasida, Wailok Tam, Taiichi Hashimoto, Motoyuki Takaai, and Eiji Aramaki. 2012. Ontoroji taiou bunpou riron to sono kousokushori no tame no conpaireson. In *Proceedings of Gengo Shori Gakkai*, Hiroshima, Japan.

Koiti Hasida. 2010. Global document annotation. `http://i-content.org/GDA`.

Japanese Gastric Cancer Association JGCA, editor. 2010. *Japanese Classification of Gastric Carcinoma*. Kanehara.

Japanese Gastric Cancer Association JGCA. 2011. Japanese classification of gastric carcinoma: 3rd english edition. In *Gastric Cancer*, pages 101–112. Springer.

Takuya Matsuzaki, Yusuke Miyao, and Junichi Tsujii. 2007. Efficient hpsg parsing with supertagging and cfg-filtering. In *Proceedings of IJCAI2007*.

Rolf Schwitter. 2010. Controlled natural languages for knowledge representation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1113–1121.