

Making Text Resources Accessible to the Reader: The Case of Patent Claims

Simon Mille¹, Leo Wanner^{1,2}

¹Pompeu Fabra University

²Institució Catalana de Recerca i Estudis Avançats
c. Ocata, 1, 08003 Barcelona

E-mail: simon.mille@upf.edu, leo.wanner@icrea.es

Abstract

Hardly any other kind of text structures is as notoriously difficult to read as patents – which is first of all due to their abstract vocabulary and their very complex syntactic constructions. Especially the claims in a patent are a challenge: in accordance with international patent writing regulations, each claim must be rendered in a single sentence. As a result, sentences with more than 200 words are not uncommon. Therefore, paraphrasing of the claims in terms the user can understand is of high demand. We present a rule-based paraphrasing module that realizes paraphrasing of patent claims in English as a rewriting task. Prior to the rewriting proper, the module implies the stages of simplification and discourse and syntactic analyses. The rewriting makes use of a full-fledged text generator and consists in a number of genuine generation tasks such as aggregation, selection of referring expressions, choice of discourse markers and syntactic generation. As generator, we use the MATE-work bench, which is based on the Meaning-Text Theory of linguistics.

1. Introduction

Hardly any other kind of text resources is as notoriously difficult to read and comprehend as patent documentation – which is first of all due to its abstract vocabulary and very complex syntactic constructions. Especially the claims in a patent are a challenge: in accordance with international patent writing regulations, each claim must be rendered in a single sentence. As a result, sentences with more than 200 words are not uncommon. Consider, for illustration, a still “rather short” claim from EP0548937:

- (1) An optical disk drive comprising: a laser light source for emitting a laser beam; an optical system for conversing the laser beam from the laser light source on a signal plane of optical disk on which signal marks are formed and for transmitting the light reflected from the signal plane; one or more optical components, arranged in the optical path between the laser light source and the optical disk, for making the distribution of the laser beam converged by the conversing means located on a ring belt just after the passage of an aperture plane of the optical system; a detection means for detecting the light reflected from the optical disk; and a signal processing circuit for generating a secondary differential signal by differentiating the signals detected by the detection means and for detecting the edge positions of the signal marks by comparing the secondary differential signal with a detection level.

A sentence of this length and complexity is difficult to process even for native speakers of English, let alone for foreigners who do not master English well. Given the enormous number of both native and non-native users reading patents on a daily basis, means that make them easier and faster to understand are of high demand. An obvious means to achieve this is their *paraphrase*, i.e., their rewriting in a more appropriate style.

In what follows, we present a rule-based module of the PATExpert service (Wanner et al., 2008) for paraphrasing of claims in patents written in English on a large scale.

The paper is structured as follows. In the next section, we present a short overview of the related work. Section 3 sketches the stages of paraphrasing as rewriting. Section 4 discusses those of these stages which are interesting from the viewpoint of generation. Section 5, finally, contains the conclusions we draw from our work and outlines some directions of future work.

2. Related work

Paraphrasing has always been considered a natural part of natural language text generation (see, among others, McKeown, 1979; Meeter & Shaked, 1988; Iordanskaja, et al., 1991; Stede, 1996; Huang & Fiedler, 1996), where it has been discussed as the problem to choose between alternative wordings (including alternative syntactic constructions) which express the same given content structure.

With the increasing popularity of web-based document retrieval, text entailment recognition, etc. the focus of the research on paraphrasing shifted considerably over the last decade in that corpus-based recognition, extraction, alignment and annotation of paraphrases became one of the main concerns (Barzilay & McKeown, 2001; Dorr et al., 2004; Marsi et al., 2007).

Our task is yet different. Given, on the one hand, the lack of paraphrased patent claim corpora and the costs to obtain such corpora, and, on the other hand, the specific features of the linguistic structures encountered in the claims (such as repetitiveness, long distance anaphoric references, etc.), we interpret claim paraphrasing as a *text regeneration*, or *rewriting*, task – with generation starting from the syntactic structure.

3. Paraphrasing as rewriting

Starting from the syntactic structure, the rewriting task presupposes a prior analysis stage, which again consists of several substages – as also does the paraphrasing stage proper.

As argued, e.g., by Iordanskaja et al. (1991), paraphrasing is more flexible and more straightforward to realize if it starts from a deep-syntactic (rather than a surface-syntactic) structure. This is also our experience. Therefore, we introduce an additional stage in which the syntactic structures derived by a parser are projected onto deep-syntactic structures. As a result, we deal with a three stage procedure, which can be depicted as follows:

1. Analysis of patent claims
 - a. Simplification, anaphoric and discourse analysis
 - b. Parsing of the simplified sentences
2. Projection of parse trees onto deep-syntactic structures
3. Paraphrasing of preprocessed patent claims
 - a. Aggregation and discourse markers
 - b. Referring expression generation
 - c. Syntactic generation

Since the preprocessing stage is discussed in detail in (Bouayad-Agha et al., submitted), we describe it in what follows only in general terms and focus in this section on stage 2. In the next section, details on stage 3, paraphrasing proper are given.

3.1 Preprocessing patent claims

The goal of the preprocessing stage is to obtain syntactic structures from which the regeneration starts. The complexity of the sentences in which the original claims are written suggests that prior to parsing, a simplification of the structure of the sentences is to be carried out. Sentence simplification and sentence compression / condensation is a popular research topic in itself (see, for instance, Clarke and Lapata, 2006). In our application, the simplification is responsible for: (i) cutting the long complex sentences of the claims into a number of simpler separate sentences taking into account surface-oriented criteria (punctuation, conjunction markers, specific cue words, specific POS-patterns, etc.); (ii) transformation of *for*-gerund constructions, which are very dominant within the linguistic style of claims; (iii) elimination of excessive anaphoric markers (such as *said ...*). Thus, for (1), the simplification returns as the first four sentences (2):

- (2) 1-An optical disk drive comprises a laser light source.
- 2-A laser light source emits a laser beam.
- 3-An optical disk drive comprises an optical system.
- 4-An optical system converses the laser beam from the laser light source on a signal plane of optical disk on which signal marks are formed. ...

During the simplification stage, also the anaphoric structure and the discourse structure (in the sense of the Rhetorical Structure Theory, Mann and Thompson, 1987) are derived. Currently, the simplification stage achieves an *f*-score of about 70% (Bouayad-Agha et al., submitted).

The result of the simplification stage is thus a sequence of simplified sentences which can be parsed with a considerably higher expectation of accuracy than the original sentence claims. For this purpose, we use the *MiniPar* dependency parser (Lin, 1998). *MiniPar* has been chosen because it produces fast and stable syntactic structures, which approximately correspond to the surface-syntactic structures in the linguistic framework underlying the generation framework we use for paraphrasing – namely the Meaning-Text Theory; see below.

The results of *MiniPar* are satisfactory, although some limitations such as systematic right-attachment can be a problem when from the syntactic structures a semantic representation or a more abstract syntactic representation needs to be derived – as, e.g., in the case of translation.

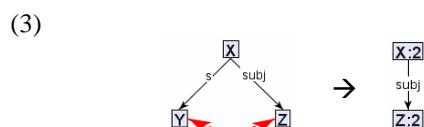
3.2 Projection onto deep-syntactic structures

All substages of paraphrasing are performed using the MATE-toolkit (Bohnet et al., 2000; Bohnet, 2006). The core of MATE is an efficient graph transducer. Although MATE can be used for any linguistic framework, it especially supports the creation and maintenance of rule-based *Meaning-Text Theory* (MTT) grammars (Mel'čuk, 1988). The linguistic model of the MTT is a multistratal model. In our scenario, the following four strata are implied: surface syntax (SSynt), deep syntax (DSynt), deep morphology (DMorph), and surface morphology (SMorph).

The syntactic structures as provided by *MiniPar* correspond to SSynt-structures. However, their vocabulary and some basic organization principles differ such that the mapping is not trivial. The SSyntSs are projected onto DSynt-structures.

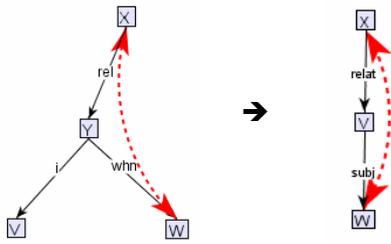
3.2.1 MiniPar-SSynt mapping

The *MiniPar* dependency structures are rather different from the SSyntSs. The mapping between the two is realized by a mapping grammar. The rules of this grammar are minimal in that each rule handles a minimal part of a *Minipar* tree. Cf., the rule in (3) which handles only the relations *s* and *subj*, mapping them onto the corresponding SSynt relation *subj*.



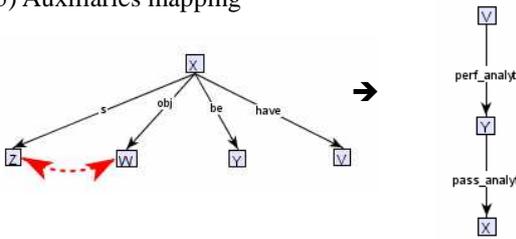
In many cases, the mapping is not straightforward, as in the case of subject relative clauses; cf. (4):

(4) Relative clause mapping



or in the case of complex verbal tense constructions; cf. (5) (the double arrow indicates co-reference):

(5) Auxiliaries mapping



It is worth mentioning that for the performance of the generator, it is crucial to limit the number of rules. In order to do so, the subject-mapping rule shown in (3) also contains strict conditions¹ that are encoded in the MATE environment, enabling this same rule to apply when any auxiliary is contained in the verb cluster. The current version of the MiniPar-SSyntS mapping grammar contains 137 rules. An evaluation of a previous version on 1324 sentences had shown that 99% of well-formed MiniPar-structures are correctly mapped onto SSyntSs.

3.2.2 SSynt-DSynt mapping

As mentioned above, the abstract nature of the DSyntS ensures more flexibility to paraphrasing. This is because of the highly abstract nature of DSyntS, which eliminates the surface-syntactic idiosyncrasies of the language in question – which is of advantage not only to paraphrasing, but also to summarization and machine translation; cf., e.g., (Mel’cuk and Wanner, 2006). Consider, for illustration, an auxiliary mapping rule in (6), where a SSynt-subtree consisting of three nodes and two arcs is mapped onto a single DSynt-node.

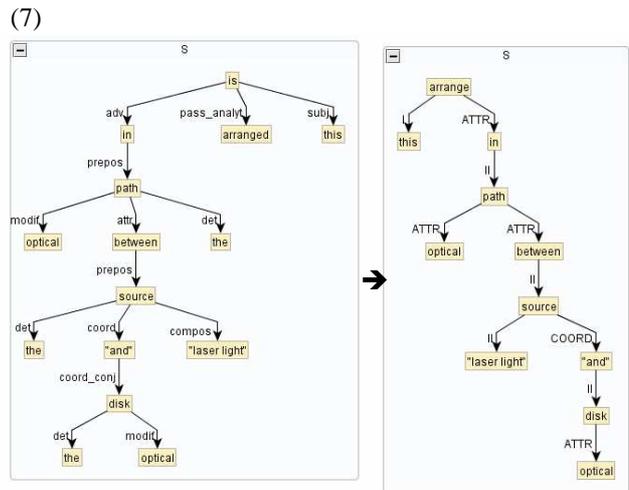


The content of the information on the DSynt node of the

¹ Each rule is assigned conditions for its application, which are not shown here.

verb V on the right hand side is the same as on the left hand side, i.e., both representations are equivalent.

During the SSyntS-DSyntS transition stage, the following four main actions are performed: (i) verbal tense auxiliary forms are mapped onto attribute-value pairs (cf. above); determiners are removed using the same strategy, i.e., they appear in DSyntS as attribute-value pairs “definiteness = DEF/INDEF...” on the node of its SSyntS governing noun; (iii) governed prepositions are removed from the structure – among them, for instance, the preposition “by” when it introduces the agent in a passive construction; (iv) some lexical units are reduced to abstract lexical labels (so-called “lexical functions”). Consider (7) for a SSyntS of a sentence from (1) -after simplification)- and its corresponding DSyntS:



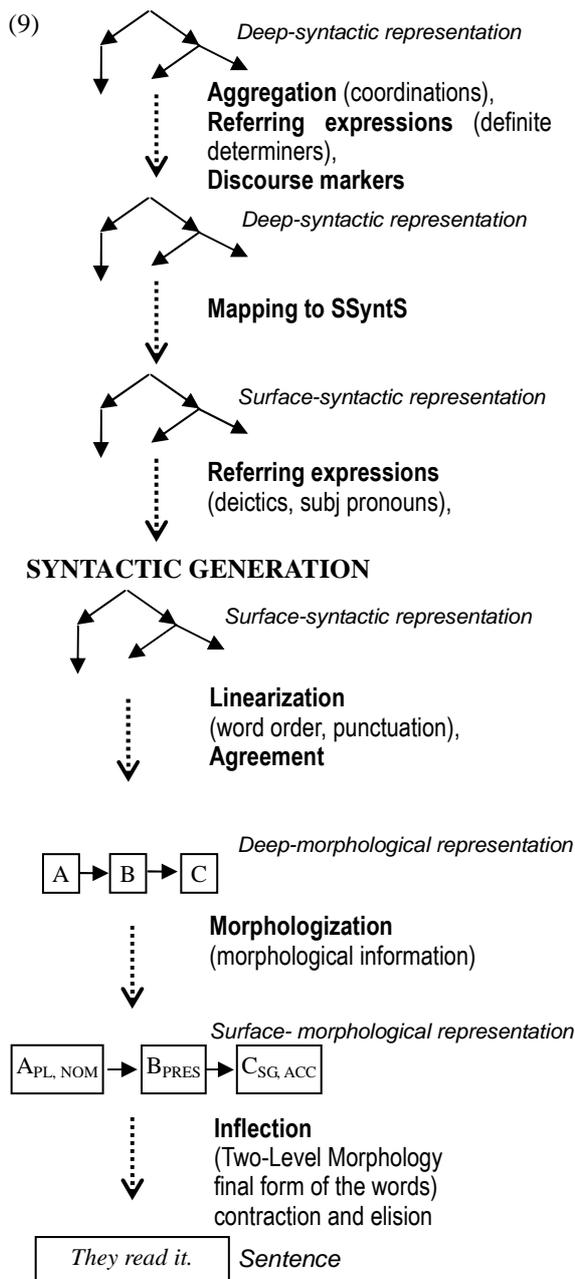
4. The process of paraphrasing

The paraphrasing procedure proper, or the regeneration, of patent claims thus starts from DSyntSs as illustrated above. The result of the regeneration conveys absolutely the same information as the original, but which is much easier to comprehend as the original. Consider, for illustration, the paraphrase of the claim (1) in (8)

- (8) An optical disk drive comprises a laser light source, an optical system, a detection means, and a signal processing circuit. Then, the laser light source emits a laser beam. The optical system converses the laser beam from the laser light source on a signal plane of optical disk. On the latter, signal mark are formed. The optical system also transmits the light reflected from the signal plane. The optical disk drive furthermore comprises one or more optical components. This is arranged in the optical path between the laser light source and the optical disk. The detection means detects the light reflected from the optical disk. The signal processing circuit generates a secondary differential signal. To do so, it differentiates the signals detected by the detection means. It also detects the edge positions of the signal mark. To do so, it compares the secondary differential signal with a detection level.

This result is obtained by traversing the three substages of paraphrasing depicted above at the beginning of Section 3. The following figure in (9) details these three substages

further and relates them to the strata in the Meaning-Text model and thus in MATE.



The substages (3a,b) of the paraphrasing procedure sketched in Section 3 are performed at DSyntS and SSyntS; surface generation (3c) is interpreted as a sequence of transitions between equivalent structures of adjacent strata: SSyntS → DMorphS → SMorphS. After the aggregation process, the DSyntS are mapped back to SSyntS for the end of the generation process; this step will not be detailed in this paper. Let us now address the three substages in more detail.

4.1 Aggregation

The simplification stage leaves us with a magnitude of simple isolated sentences which need to be aggregated. “Aggregation” is the fusion, by means of syntactic

coordination, of several separate sentence or phrase structures that share common parts into one structure in which the previously common parts occur only once (Dalianis, 1996). The main criterion for allowing two sentences to be aggregated is checking the co-reference of their components.

In our application, two main kinds of aggregation are distinguished: subject aggregation and object aggregation.

4.2.1 Subject Aggregation: Object coordination

The following rule is applied to a part of the simplified main claim in (2):

$$(10) \quad [X(\text{id}=\text{n}) Y_{\text{verb}} Z_1] + [X(\text{id}=\text{n}) Y_{\text{verb}} Z_2] + \dots + [X(\text{id}=\text{n}) Y_{\text{verb}} Z_n] \\ \rightarrow \\ X Y_{\text{verb}} [Z_1 \text{ and } Z_2 \text{ and } \dots \text{ and } Z_n]$$

$$(11) \quad {}^1[An \text{ optical disk drive}]_X [comprises]_Y [a \text{ laser light source}]_{Z1}. \\ {}^3[An \text{ optical disk drive}]_X [comprises]_Y [an \text{ optical system}]_{Z2}. \\ \rightarrow \\ (1+3) An \text{ optical disk drive comprises a laser light source and an optical system.}$$

In case there are more than two sentences to be aggregated, all non-final objects are separated by commas.

This kind of aggregation is licensed if X is a subject, Z_i any kind of object. I is limited to the “comprise”-like verbs – for instance “include” and “form”, which are very frequent in the patent genre.

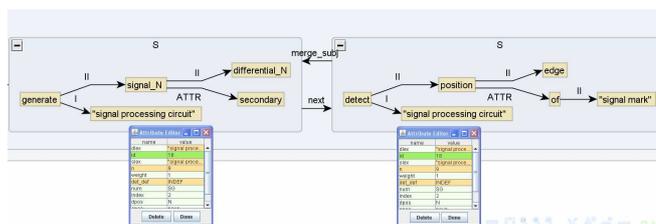
4.2.2 Subject aggregation: Verb coordination

Another type of subject aggregation addresses verb coordination:

$$(12) \quad [X(\text{id}=\text{n}) Y_{1\text{verb}} Z_1] + [X(\text{id}=\text{n}) Y_{2\text{verb}} Z_2] + \dots \\ + [X(\text{id}=\text{n}) Y_{3\text{verb}} Z_n] \\ \rightarrow \\ X [Y_{1\text{verb}} Z_1, Y_{2\text{verb}} Z_2, Y_{3\text{verb}} Z_3, \dots \text{ and } Y_n\text{verb} Z_n]$$

Two sentences have the same subject (with the coreference being identified by the attribute “id=18” on each node) and different main verbs:

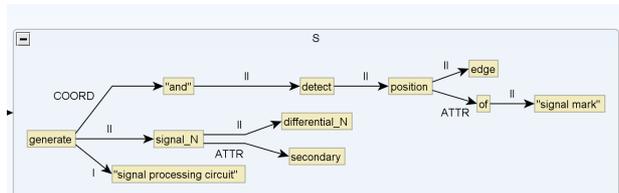
$$(13)$$



One of the main conditions of application of this rule is that the second sentence is not too long, so as not to build huge sentences; its deep-syntactic weight must be inferior

to 11 nodes. Furthermore, the “comprise”-like verbs are excluded from this rule, as well as the verb “be” (see next subsection). Cf. (14) for illustration:

(14)



4.2.3 Subject aggregation: “BE” coordination

The case of the copula –frequently used in patents as shown in the left side of the rule below- is isolated from the other verbs since the conjunction used for aggregation is not the same:

(15)

$[X(\text{id}=n) \text{ BE } Z_1] + [X(\text{id}=n) \text{ BE } Z_2] + \dots [X(\text{id}=n) \text{ BE } Z_n]$
 \rightarrow
 $X \text{ BE } [Z_1, Z_2, \text{ or } Z_n]$

(16)

The optical component is a shading member. + The optical component is a transparent conical body.
 \rightarrow
The optical component is a shading member or a transparent conical body.

4.2.4 Object aggregation: Introduction of relative clauses

The last type of aggregation is handled in SSynt, along with the processing of referring expressions: if the subject of a verb is the same as the object in the previous sentence, a relative clause is introduced:

(17)

$[X Y_{1\text{verb}} Z_1(\text{id}=n)] + [Z_1(\text{id}=n) Y_{1\text{verb}} Z_2] \cap (Y_1.\text{weight} + Y_2.\text{weight})=\text{light}$
 \rightarrow
 $X [Y_{1\text{verb}} Z_1, \text{ which/that } Y_{1\text{verb}} Z_2]$

(18)

A disk device comprises the disk tray. The disk tray comprises a guide part.
 \rightarrow
A disk device comprises the disk tray, which comprises a guide part.

One condition for the application of this rule is that the phrase to become the relative clause and its matrix clause are syntactically not too “heavy”, because we want to keep the sentences relatively short. If the conjunction of sentences is too heavy, the introduction of a deictic is preferred. This rule does not apply either if Z_1 is already aggregated so as not to rebuild sentences that would be very long, as in (1).

4.3 Adding Discourse Markers

In order to keep the semantic links between the sentences after simplification, some rules add discourse markers to the top verb of the DSyntS. Depending on the discourse relation which is introduced during the simplification stage, the marker can be retrieved from a discourse-marker dictionary. For instance, consider the following extract of (1):

(19)

*... a signal processing circuit **for** generating a secondary differential signal by differentiating the signals detected by the detection means.*

The simplification stage provides us two simplified sentences linked with a discursive relation “means”, corresponding to the syntactic marker “by+Ving”:

(20)

[A signal processing circuit generates a secondary differential signal] –means-> [A signal processing circuit differentiates the signals detected by the detection means.]

In MATE, the marker corresponding to the “means” relation is retrieved and introduced in the deep-syntactic structure:

(21)

*A signal processing circuit generates a secondary differential signal. **To do so**, a signal processing circuit differentiates the signals detected by the detection means.*

If there are several markers for one relation, MATE is able to provide as several output structures, one of which only will go through the rest of the generation. For instance, “to do so” is sometimes realized as “for this”, in order to avoid systematic repetitions in the paraphrased text. Furthermore, in order to improve the readability of the aggregated text, various adverbs are introduced in the DSyntS:

- If a sentence with the same top verb and the same subject as a previous sentence has not been aggregated, “furthermore” is added:

(22)

*An optical disk drive comprises a laser light source, an optical system, a detection means, and a signal processing circuit. [...] An optical disk drive **furthermore** comprises one or more optical components.*

- If two consecutive sentences have the same subject and not the same verb, “in addition” is introduced as a modifier of the second top verb:

(23)

*A signal processing circuit generates a secondary differential signal. **In addition**, a signal processing circuit detects the edge positions of the signal mark.*

- If two non consecutive sentences have the same subject and not the same verb, “also” is this time

introduced as a modifier of the second top verb:

(24)

*A signal processing circuit generates a secondary differential signal. [...] A signal processing circuit **also** detects the edge positions of the signal mark.*

4.4 Referring Expression Generation

The sentences (21-24) are the way they would be generated if there was no further process. It is obvious that the introduction of referring expressions is crucial to improve the overall quality of the paraphrases. At the moment, three types of referring expressions are handled by our grammars: the definite article, deictic determiner, and subject pronoun. The algorithm used for co-reference resolution, which enables the processing of referring expressions, is simply based on the claim structure of the patent. Every time a nominal group appears identically within a group of dependent claims, it is given the same attribute "id=n". This is based on the assumption that in a patent, every time a particular noun appears, it refers to the exact same entity, as long as we remain within the same conjunct of dependent claims.

4.4.1 Definite determiners

In the case of (22), a simple rule introduces an attribute "definiteness=DEFINITE" on every DSyntS nominal node the "id" attribute of which has been previously encountered in the structure. The result on the surface level of the application of this rule is the following:

(25)

*An optical disk drive comprises a laser light source, an optical system, a detection means, and a signal processing circuit. [...] **The** optical disk drive furthermore comprises one or more optical components.*

4.4.2 Deictics

The rule introducing the deictic determiners has almost the same left side as the one for relatives as seen in (17). It applies –on the SSyntS- when (17) does not apply, which is when the weight of the conjunct ($Y_1 + Y_2$) does not exceed 30 SSynt nodes, i.e. 30 lexical units including determiners, auxiliaries, etc.

(26)

$[X Y_{1\text{verb}} Z_1(\text{id}=n)] + [Z_1(\text{id}=n) Y_{1\text{verb}} Z_2] \cap (Y_1.\text{weight} + Y_2.\text{weight})=\text{heavy}$

→

$X Y_{1\text{verb}} Z_1. \text{This } Z_1 Y_{2\text{verb}} Z_2.$

(27)

*An optical disk drive comprises a laser light source, an optical system, a detection means, and a signal processing circuit. **This** signal processing circuit is shaded by is a shading member arranged near the optical axis around the aperture plane of the optical system or by a transparent conical body arranged near the optical axis around the aperture plane of the optical system.*

4.4.3 Pronoun subject

The last type of configuration we want generate a referring expression for is when two sentences have the same subject but not the same top verb, as it is the case in (21). All concerned subjects should have the same value for their "id" attribute and be in consecutive sentences:

(28)

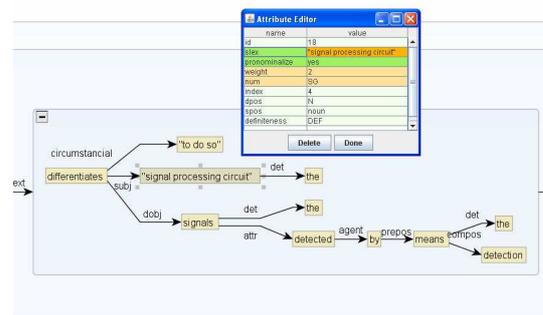
$[X_1(\text{id}=n) A_{\text{verb}} Z_1] + [X_2(\text{id}=n) B_{\text{verb}} Z_2] + \dots + [X_n(\text{id}=n) C_{\text{verb}} Z_n]$

→

$X A_{\text{verb}} Z_1 + \text{it } B_{\text{verb}} Z_2 + \text{it } C_{\text{verb}} Z_n]$

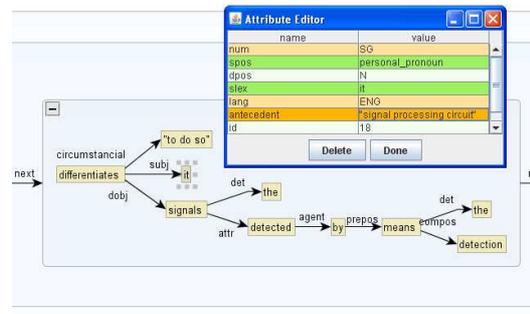
In a first step, every "X" from X_2 to X_n is marked for pronominalization, as shown in the following figure:

(29)



Then, a personal pronoun is introduced in the structure; the original number is kept, and a trace of the antecedent as well so as to be able to retrieve its gender from the lexicon:

(30)



Hence, (21) becomes (31):

(31)

*A signal processing circuit generates a secondary differential signal. To do so, **it** differentiates the signals detected by the detection means.*

4.5 Syntactic generation

As indicated in Figure (9), syntactic generation starts with the SSynt–DMorph transition. This transition involves three types of rules: word order, agreement and punctuation. Word order rules are further divided into two types: vertical order rules and horizontal order rules. The first specify the relative order between a governor and one of its dependents. They are sensitive to the kind of syntactic relation between the words as well as to any

features the said words may have. Cf. the rule for the definition of the order between the subject and the verb from which it depends:

$$(32) \quad X \text{—subject—} \rightarrow Y \Rightarrow Y < X$$

The second type of rules specifies the relative order between two (or more) dependents of a same governor. The following rule states that the subject goes before any other dependent of its governor, except circumstantials:

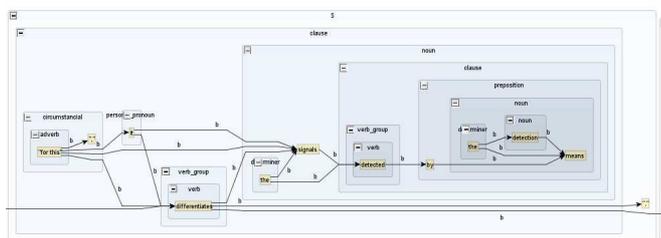
$$(33) \quad X \text{—subject—} \rightarrow Y \wedge X \text{—}r\text{—} \rightarrow Z \Rightarrow Y < Z \mid r \neq \text{circumstantial}$$

Agreement rules recopy grammatical information from one node to another. Thus, the number and person of the subject are recopied to the verb from which it depends; cf.:

$$(34) \quad X \text{—subject—} \rightarrow Y \Rightarrow X.\text{person} = Y.\text{person} \wedge X.\text{number} = Y.\text{number}$$

Punctuation rules are rules that insert commas after circumstantials, periods at the end of a sentence, and additional markers that make the text more readable. Consider an example of a DMorphS, corresponding to the SSyntS seen in (30):

(35)



Errors can occur in any of the modules mentioned in the context of paraphrasing and generation. Therefore, a fallback strategy is needed in order to avoid that information is lost. At any step during the paraphrasing and generation, if an error is detected, the erroneous structure is replaced by a simple fallback structure containing the original simplified sentence before parsing.

5. Conclusions

The described paraphrasing strategy has been tested on about 500 sentences. The first evaluation round with human experts has shown that the module delivers accurate paraphrasing in 95%. The accuracy is also due to the fallback strategy that has been defined so as to avoid the generation of ill-formed sentences: filtering rules prevent the majority of them to be generated; instead, the original sentence before parsing is included as canned text into the SSynt-representation.

So far, the corpus of patent claims we worked with was

compiled from two technological areas: optical recording devices and machine tools. Future work will include further extension of the grammars and lexica with the goal to broaden the coverage of the technological areas. Further work will include a revision of the techniques involved in the preprocessing stage so as to improve the quality of the raw material the paraphrasing stage proper starts from.

6. Acknowledgements

The work described in this paper has been carried out in the framework of the PATExpert project funded by the European Commission under contract number FP6 028116. Many thanks to our colleagues and friends from the TALN-group at UPF, Stefan Bott, Nadjet Bouayad-Agha, Gerard Casamayor, Gaby Ferraro, Johannes Graën, Anna Joan, and Vanesa Vidal, for their support. We would also like to thank the three LREC-reviewers for their valuable comments which significantly improved the quality of the final version of the paper.

7. References

- Barzilay, R. & Lee, L. (2003). Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT*, pp. 16–23.
- Bohnet, B. (2006). Textgenerierung durch Transduktion linguistischer Strukturen. DISKI 298. AKA: Berlin.
- Bohnet, B., Langjahr, A. & Wanner, L. (2000). A development environment for an MTT-based sentence generator. In: *Proceedings of the First International Conference on Natural Language Generation*, Mitzpe Ramon, Israel, pp. 260–263.
- Bouayad-Agha, N., G. Ferraro, V. Vidal and L.Wanner. 2007. “The first steps towards a simplification system of patent’s claims. Description and evaluation.” Submitted.
- Clarke, J. and Lapata, M. (2006). Models for sentence compression: a comparison across domains, training requirements and evaluation measures. In *Proceedings of COLING and ACL*, pp. 377-384.
- Dalianis, H. (1996). Concise Natural Language Generation from Formal Specifications. PhD Thesis. Department of Computer and System Sciences. Royal Institute of Technology. Stockholm.
- Dorr, B., Green, R., Levin, L., Rambow, O., Farwell, D., Habash, N., Helmreich, S., Hovy, E., Miller, K.J., Mitamura, T., Reeder, F., & Siddharthan, A. (2004). Semantic Annotation and Lexico-Syntactic Paraphrase. In *Proceedings of LREC*.
- Dras, M. (1999). *Tree Adjoining Grammar and the Reluctant Paraphrasing of Text*. Ph.D. thesis, Macquarie University.
- Huang, X. and A. Fiedler. (1996). Paraphrasing and Aggregating Argumentative Text Using Text Structure. In *Proceedings of the 8th International Workshop on Natural Language Generation*. Pp. 21-30. Herstonceux Castle, UK.
- Iordanskaja, L. R. Kittredge and A. Polguère. (1991). Lexical selection and paraphrase in a Meaning-Text generation model. In C. Paris, W. Swartout, and W. Mann (eds.). *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer:

Dordrecht.

- Lin D. (1998). Dependency-based Evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems*.
- Mann, W. and S. Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. Technical report, Information Science Institute (ISI), University of Southern California, Los Angeles.
- Mel'cuk, I. and L. Wanner. (2006). Syntactic Mismatches in Machine Translation. In *Machine Translation* 20(2), 81-138.
- Marsi, E., E. Krahmer, and W. Bousma. (2007). Dependency-based paraphrasing for recognizing textual entailment. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, held in conjunction with the Annual Meeting of the ACL, Prague.
- McKeown, K.R. (1979). Paraphrasing using given and new information in a question-answer system. In *Proceedings of the ACL*, pp. 67-72.
- Meeter, M. and V. Shaked. (1988). Strategies for effective paraphrasing in *Proceedings of COLING*, 431-436.
- Stede, M. (1996). Lexical paraphrases in multilingual sentence generation. In *Machine Translation* 11:75-107.
- Wanner, L. et al. (2008). PATExpert: Towards Content-Oriented Patent Document Processing In *World Patent Information Journal*, 30(1):21-33.