

# Reversing Controlled Document Authoring to Normalize Documents

Aurélien Max

Groupe d'Etude pour la Traduction Automatique (GETA)

Xerox Research Centre Europe (XRCE)

Grenoble, France

aurelien.max@imag.fr

## Abstract

This paper introduces document normalization, and addresses the issue of whether controlled document authoring systems can be used in a reverse mode to normalize legacy documents. A paradigm for deep content analysis using such a system is proposed, and an architecture for a document normalization system is described.

## 1 Introduction

Controlled Document Authoring is a field of research in NLP that is concerned with the interactive production of documents in limited domains. The aim of systems implementing controlled document authoring is to allow the user to specify an underlying semantic representation of the document that is well-formed and complete relative to its class of documents. This representation is then used to produce a fully controlled version of the document, possibly in several languages. We distinguish controlled document authoring systems from what is referred to in (Reiter and Dale, 2000) as *computer as authoring aid*, which are Natural Language Generation systems intended to produce initial drafts or only routine factual sections of documents, in that the former can be used to produce high-quality final versions of documents without the need for further hand-editing.

The question which motivated our work was the following: can we reuse the resources of an existing controlled document authoring system to

analyze documents from the same class of documents? If so, we could obtain the semantic structure corresponding to a raw document, and then produce from it a completely controlled version. If the raw document is bigger in scope from the documents that the authoring system models, then something similar to document summarization by content recognition and reformulation would be done. Incomplete representations after automatic analysis could be interactively completed, thus re-entering controlled document authoring. Producing the document from the semantic representation in several languages would do some kind of *normalizing translation* of the original document (Max, 2003). We call the process of reconstructing such a semantic representation (and re-generating controlled text in the same language), which is common to all the above cases, *document normalization*.

In this paper, we will first attempt to argue why document normalization could be of some use in the real world. We will then introduce our approach to document normalization, and describe a possible implementation. We will conclude by introducing our future work.

## 2 Why normalize documents?

Text normalization often refers to techniques used to disambiguate text to facilitate its analysis (Mikheev, 2000). The definition for document normalization that we propose can have much more impact on the surface form of documents.

In order to propose application domains for document normalization, we attempted to identify do-

mains where documents of the same nature but from different origins were compiled into homogeneous collections. We focussed our attention on the pharmaceutical domain, which produces several yearly compendiums of drug leaflets, as for example the French Vidal de la Famille (OVP Editions du VIDAL, 1998).

Producing pharmaceutical documents is the responsibility of the pharmaceutical companies which market the drugs. A study we conducted on a corpus of 50 patient pharmaceutical leaflets for pain relievers (Max, 2002) collected on drug vendor websites revealed several types of variations. The first observation was that the structures of the leaflets could vary considerably. For comparable drugs, we found that for example warning-related information could be presented in different ways. One of them was to divide them into two sections, *Warnings* and *Side effects*, another one had a three-section division into *Drug interaction precautions*, *Warnings*, and *Alcohol warning*. In the first case, drug interaction precautions effectively appeared in the more general *Warnings* section (*You should ask your doctor before taking aspirin if you are taking medicines for...*) Conversely, possible side effects, which are in a separate section in the first case, were found in the *Warnings* section in the second case (*If ringing in the ears or a loss of hearing occurs...*) A related type of variation concerns the focus which is given to certain types of content. A warning specific to alcohol is needed for patient taking aspirin as alcohol consumption may cause stomach bleeding in this circumstance. While some leaflets presented a separate section, *Alcohol warnings*, others simply mentioned the related possible side effect, stomach bleeding, in the appropriate section.

In spite of these differences in structure, leaflets in the subset we have studied usually express the same types of content, that is, the communicative intentions expressed by the authors of the leaflets are similar. However, this content can be expressed in a variety of ways. A factor analysis of stylistic variation in a corpus of 342 patient leaflets (Paiva, 2000) revealed that two important factors opposed *abstraction* (e.g. use of agentless passives and nominalizations) to *involvement/directness* (e.g. use of 1st and 2nd

persons and imperatives) and *full reference* to *pronominalized reference*.

Our study also showed that similar communicative intentions could be expressed in a variety of ways conveying more or less subtle semantic distinctions. We argue that for documents of such an important nature, consistency of expression and of information presentation can not only be beneficial to the reader but also necessary to allow a clear and unambiguous understanding of the communicative intentions contained in different documents. Controlled document authoring systems can guarantee that the documents they produce are consistent as the production of the text is under the control of the system. An authoring system for drug leaflets conforming to Le Vidal specifications has been developed (Brun et al., 2000), showing that new documents could be written in a fully controlled way. But most existing documents, if they conform to some specifications, do not have these desirable properties across different drug vendors. Our research thus addresses this complementary issue: can we reuse the modelling of documents of such systems to analyze existing *legacy documents* from the same class of documents?

Document normalization implies analyzing a legacy document into a semantically well-formed content representation, and producing a normalized version from that content representation. This expresses *predefined communicative content* present in the input document, in a structurally and linguistically controlled way. Predefined content reveals communicative intentions, which should ideally be described by an expert of the discourse domain.

### 3 Controlled document authoring

There has been a recent trend to investigate controlled document authoring, e.g. (Power and Scott, 1998; Dymetman et al., 2000), where the focus is on obtaining document content representations by interaction with the user/author and producing multilingual versions of the final document from them. Typically, the user of these systems has to select possible semantic choices in active fields present in the evolving text of the document in the user's language. These selections iteratively refine

```

listOfProductWarnings(AllergyWarning, DurationWarning)::productWarnings(InstanceOfSymptom, ActiveIngredient)-e --->
  ['PRODUCT WARNINGS'],
  AllergyWarning::allergyWarning(ActiveIngredient)-e,
  DurationWarning::durationWarning(InstanceOfSymptom)-e.
doNotTakeInCaseOfAllergy(Ingredient)::allergyWarning(Ingredient)-e --->
  ['DO NOT TAKE THIS DRUG IF YOU ARE ALLERGIC TO '],
  Ingredient::activeIngredient-e.
doNotTakeForMore(Number, TimeUnit, PWWarning)::durationWarning(InstanceOfSymptom) --->
  [' This product should not be taken for more than '],
  Number::integer-e, TimeUnit::timeUnit-e,
  [' without consulting a doctor. '],
  PWWarning::persistOrWorsenWarning(InstanceOfSymptom).
consultIfPainPersistsOrGetsWorse::persistOrWorsenWarning(pain) --->
  ['Consult your doctor if pain persists or gets worse.'].

```

Figure 1: MDA grammar extract for the product warning section of a patient drug leaflet.

the document content until it is complete.

In the Multilingual Document Authoring (MDA) system (Dymetman et al., 2000), the specification of well-formed document content representations can be recursively described in a grammar formalism that is a variant of Definite Clause Grammars (Pereira and Warren, 1980). Figure 1 shows a simple MDA grammar extract for the product warning section of a patient leaflet. The first rule reads as follows: the semantic structure `listOfProductWarning(AllergyWarning, DurationWarning)` is of type `productWarnings(InstanceOfSymptom, ActiveIngredient)`, and is made up of the terminal string “PRODUCT WARNINGS”, an element of type `allergyWarning(ActiveIngredient)` element, and an element of type `durationWarning(InstanceOfSymptom)`. Semantic constraints are established through the use of shared type-parameters: for example, `InstanceOfSymptom` constrains the element of type `durationWarning`. Text strings can appear in right-hand sides of rules, which allows to associate text realizations to content representations by a traversing of the leaves of their tree.<sup>1</sup> The granularity of text fragments that is allowed in rules is not necessarily as fine-grained as predicate-argument structures of sentences commonly used in NLG. This approach proved to be adequate for classes of documents where certain choices could be rendered as entire text passages (e.g. pregnancy warnings, disclaimers, etc.) and where

<sup>1</sup>Some details such as morphological-level constraints have been omitted for lack of space.

a more fine-grained representation would not be needed, thus offering an interesting intermediate level between full NLG and templates (Reiter, 1995).

## 4 A paradigm for deep content analysis

### 4.1 Fuzzy inverted generation

Content analysis is often viewed as a parsing process where semantic interpretation is derived from syntactic structures (Allen, 1995). In practice, however, building broad-coverage syntactically-driven parsing grammars that are robust to the variation in the input is a very difficult task. Furthermore, we have already argued that for the purpose of document normalization we would like to match texts that do not carry significant communicative differences in a given class of documents but may be of quite different surface forms. Therefore, we propose to concentrate on what counts as a well-formed document semantic representation rather than on surface properties of text, as the space of possible content representations is vastly more restricted than the space of possible texts.

Bridging the gap between deep content and surface text can be done by using the textual predictions made by the generator of an MDA system from well-formed content representations. Indeed, an MDA system can be used as a device for enumerating well-formed document representations in a constrained domain and associating texts with them. If we can compute a relevant measure of semantic similarity between the text produced for any document content representation

and the text of a legacy document, we could possibly consider the representations with the best similarity scores as those best corresponding to the legacy document under analysis. As this kind of analysis uses predictions made by a natural language generator, we named it *inverted generation* (Max and Dymetman, 2002). And because a generator will seriously undergenerate with respect to all the texts that could be normalized to the same communicative intention, we made this process *fuzzy* by matching documents at a more abstract level than on raw text to evaluate commonality of communicative content.

#### 4.2 Implementing fuzzy inverted generation using MDA

We use the formalism of the MDA authoring system (Dymetman et al., 2000) to implement fuzzy inverted generation, as it offers a close coupling between semantic modelling and text generation.<sup>2</sup> In this context, an input document will be used as an information source to reconstruct the semantic choices that *a human author would have made if she had created the document most similar to the input document in terms of communicative content using MDA*. The space of virtual documents<sup>3</sup> for a given class of documents being potentially huge, we will want to implement a heuristic search procedure to find the best candidates. The confidence in the analysis will depend on the quality of the match and the similarity measure used, which suggests that in practice such a normalization task could hardly be done without at least some intervention from a human expert.

The search for candidate content representations begins under the assumption that the input document belongs to the class of documents modeled by the MDA grammar used. Starting from the root type of the MDA grammar, partial content representations are iteratively produced by performing steps of derivation on the typed abstract trees. This corresponds to instantiating a variable with a value compatible with its type (which is

<sup>2</sup>MDA grammars being Prolog programs, any Prolog predicate could be called from the rules. We ignored this powerful feature of MDA grammars and thus used a simplified formalism.

<sup>3</sup>We call *virtual documents* documents that can be predicted by the semantic model but do not exist *a priori*.

what is done interactively in the authoring mode). A similarity measure is computed between the input document and the *set of all the virtual documents that could be produced from a given partial content representation*. This similarity measure is used as the evaluation function of an admissible heuristic search (Nilsson, 1998) that returns the candidate content representations in decreasing order of similarity with the input document. In order to guarantee that the search is admissible, it has to implement a best-first strategy, and use an *optimistic* evaluation function that decreases as search progresses and that is an overestimate of the similarity between the best attainable virtual document and the input document.

In order to allow the computation of the similarity function between a partial content representation (a node in our search space) and an input document, some account of the properties of attainable virtual documents has to be percolated to the semantic types in the grammar. We call *profile* a representation of a text document that can be used to measure some semantic content similarity. A profile must have the property that it can be computed for text strings appearing in rules of the MDA grammar and percolated to semantic types in the grammar up to the root type. A profile for a type gives an account of the profiles of all the terminals attainable from it, in such a way that the similarity function used will overestimate the value of the similarity between the best attainable virtual document and the input document. We will show in the next section how this can be realized in a practical normalization system using an MDA grammar.

## 5 A possible implementation of a document normalization system

### 5.1 System architecture

In this section we describe the architecture of the document normalization system that we have started to develop. An MDA grammar is first compiled to associate profiles with all its semantic types. This compiled version of the grammar is used in conjunction with the profile computed for the input document in a first pass analysis. The aim of this first pass analysis, implementing

fuzzy inverted generation, is to isolate a limited set of candidate content representations. A second pass analysis is then applied on those candidates, which are then actual texts associated with their content representation. Ultimately, interactive disambiguation takes place to select the best candidate among those that could not be filtered out automatically.

## 5.2 Profile construction

**Profile definition** Profiles give an account of text content and are compared to evaluate content similarity. We defined our notion of content similarity from the fact, broadly accepted in the information retrieval community, that the more terms (and related terms) are shared by two texts, the more likely they are to be about the same topic. Text content can be roughly approximated by a vector containing all lemmatized forms of words and their associated number of occurrences. We call such a vector the *lexical profile* of a text. It has been shown that using sets of synonyms instead of word forms could improve similarity measures (Gonzalo et al., 1998), so we use *synset profiles* to account for lexico-semantic variation.

**Text profile construction** Words in text fragments are first lemmatized and their part-of-speech is disambiguated using the morphological analysis tools of XRCE. Their corresponding set of synonyms is then looked up through a lexico-semantic interface, and the corresponding synset key is used to index the word or expression. We have developed an annotation graphical interface that allows a human to annotate strings in MDA grammars by choosing the appropriate synset in the default lexico-semantic resource, WordNet (Miller et al., 1993), or to define new sets of synonyms in the absence of availability of a more specific resource. The annotation interface also allows the annotator to specify a value of *informativity* for the indexed synsets<sup>4</sup>, which is taken into account when computing profile similarity. The set of synsets which have been used to

<sup>4</sup>We thought that the kind of informativity for words that was needed required some expertise on the class of documents, and was therefore not easily derivable from corpus statistics. We nevertheless intend to evaluate informativity measures derived from term frequencies.

index the text fragments found in the MDA grammar is then used as a target set when building the profile for an input document.

**Profile similarity computation** We want to evaluate how much content is common to an input document and a set of virtual documents, but for our purpose we do not want this measure to be penalized by unshared content. Furthermore, we want to use this measure as the evaluation function of our search procedure, so it has to be optimistic when applied to partial representations. Thus we chose a simple intersection measure between two lexical profiles, weighted by the informativity of the synsets involved. This measure is given by the following formula, where  $occs_{P1}(item)$  is the number of occurrences of *item* in profile *P1*, and  $inf(item)$  is its informativity:

$$sim(P1, P2) = \sum_{item \in P1, P2} \frac{\min(occs_{P1}(item), occs_{P2}(item)) * inf(item)}{occs_{P2}(item)}$$

## 5.3 Grammar precompilation

A given semantic type can have several realizations, which correspond to a collection of virtual texts. The synset profile of a type has to give an account of the maximum number of occurrences of elements from a synset that can be obtained by deriving this type in any possible way. The synset profile for an expansion of a type (a right-hand side of a rule) can be obtained by taking the bag-union (which sums the number of occurrences for each element in the profiles) of the synset profiles of all the elements in the expansion. Obtaining the profile for a type can then be done by taking the *maximum* of the profiles of all its expansions. We call this operation, which takes for each element its maximum number of occurrences in the expansions of the type, the *union-max* of the profiles of all the expansions for a type. This reflects the fact that, whatever the derivation that is made from a type, elements from a given synset cannot appear in a text produced from that derivation more than a given number of times.

The grammar precompilation algorithm shown on figure 2 uses a fixpoint approach. At each iteration, the profiles for all the semantic types are

```

currentIteration <- 0
maximumNumberOfIterations <- number of semantic types in the grammar
thereWasAnUpdate <- true
create an empty profile for every semantic type
REPEAT WHILE thereWasAnUpdate is true AND currentIteration <= maximumNumberOfIterations
  FOR ALL semantic types in the grammar
    FOR ALL their expansions
      build the profile for that expansion given the current profiles
      set the profile for that type to be the union-max of itself and the profile for the expansion
      IF (currentIteration = maximumNumberOfIterations)
        set all changing numbers of occurrences for elements in the profiles to an infinity value
      update thereWasAnUpdate appropriately
      currentIteration <- currentIteration + 1

```

Figure 2: Algorithm for percolating profiles in the grammar

built, given the current values of the profiles involved in their construction. If no profile update has been done during an iteration, then a fixpoint has been reached and all the synset elements have been percolated up to the root semantic type. If updates are still made after a certain number of iterations, which corresponds to the number of semantic types in the grammar, that is, the depth of the longest derivation without repetition, then the corresponding updated values will tend to infinity (this corresponds to the case of recursive types).

#### 5.4 Automatic selection of candidates

A first pass analysis implements fuzzy inverted generation. The most promising candidate content representations are expanded first. Their profile is the bag-union of the profiles for the types of all their uninstantiated variables (the unspecified parts) and the profiles for their text fragments (the known parts). The intersection similarity measure can only decrease or remain constant as a partial content representation is further refined, thus satisfying the constraint for the admissibility of the search. The search terminates when a given number of complete candidates have been found.<sup>5</sup>

This first pass restricts the search space from a huge collection of virtual documents to a comparatively smaller number of concrete textual documents, associated with their semantic structure. Candidates differ in at least one semantic choice, so the various alternatives can be rescored lo-

<sup>5</sup>This number has to be determined empirically for a given source of documents so that it guarantees that the correct candidate is retained.

cally using more fine-grained measures. An approach can be to search for evidence of the presence of some text passages produced by competing semantic choices in the input document, and to rescore them appropriately. Given the constraints on the domain of the input documents, we hope that simple features will help significantly in disambiguating candidates, as for example distance constraints which have been shown to participate significantly in the evaluation of text similarity over short passages (Hatzivassiloglou et al., 1999).

#### 5.5 Interactive disambiguation

Due to its limitations, the proposed approach cannot guarantee that the correct candidate can be selected automatically. Recognizing similar communicative intentions challenges simple text matching techniques, and can require expert knowledge that is difficult to obtain *a priori* and to encode into automatic disambiguation rules. We therefore propose that automatic selection of candidates be done down to a level of confidence that would be determined so as to guarantee that the correct document is retained. We then envisage several modes of intervention from an expert. One is to display the texts corresponding to possible alternatives among which the expert could select the correct one in the light of highlighted passages of the document that obtained good scores during the second pass analysis. Supervised learning of new formulations could then be done by allowing the expert to augment the generative power of the MDA grammar used by adding alternative termi-

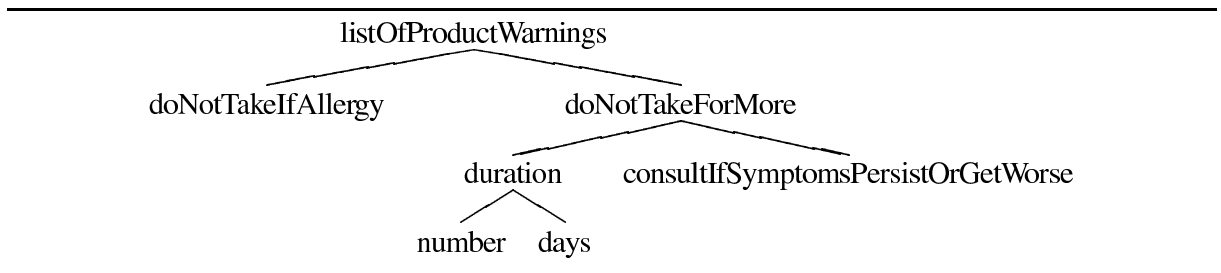


Figure 3: The abstract tree for the product warning section of the normalized document on figure 4

<p><b>Indications:</b> For the temporary relief of minor aches and pains associated with headache, a cold, muscular aches, backache, toothache, premenstrual and menstrual cramps and the minor pain of arthritis.</p> <p><b>Directions:</b> <b>Adults and children 12 years of age and older:</b> 2 tablets with a full glass of water every 6 hours while symptoms persist, or as directed by a doctor. Do not exceed 8 tablets in 24 hours. <b>Do not give to children under 12 years of age.</b></p> <p><b>Ingredients:</b> <b>Active Ingredients:</b> Each tablet contains: Aspirin (500 mg), Caffeine (32 mg) <b>Inactive Ingredients:</b> Hydroxypropyl Methylcellulose, Microcrystalline Cellulose, Polyethylene Glycol, Sodium Lauryl Sulfate, Starch</p> <p><b>Drug Interaction Precautions:</b> Do not take this product if you are taking a prescription drug for anticoagulation (thinning the blood), diabetes, gout or arthritis unless directed by a doctor.</p> <p><b>Warnings:</b> <b>Children and teenagers should not use this medicine for chicken pox or flu symptoms before a doctor is consulted about Reye syndrome, a rare but serious illness reported to be associated with aspirin. Do not take this product if you have asthma, an allergy to aspirin, stomach problems (such as heartburn, upset stomach, or stomach pain) that persist or recur, ulcers or bleeding problems, or if ringing in the ears or a loss of hearing occurs, unless directed by a doctor. Do not take this product for pain for more than 10 days unless directed by a doctor. If pain persists or gets worse, if new symptoms occur, or if redness or swelling is present, consult a doctor because these could be signs of a serious condition. As with any drug. If you are pregnant or nursing a baby, seek the advice of a health professional before using this product. It is especially important not to use aspirin during the last 3 months of pregnancy unless specifically directed to do so by a doctor because it may cause problems in the unborn child or complications during delivery.</b> Keep this and all drugs out of the reach of children. In case of accidental overdose, seek professional assistance or contact a poison control center immediately.</p> <p><b>Alcohol Warning:</b> If you consume 3 or more alcoholic drinks every day, ask your doctor whether you should take aspirin or other pain relievers or fever reducers. Aspirin may cause stomach bleeding.</p>	<p><b>INDICATIONS</b> For the temporary relief of minor aches and pains, including: - muscular aches - headaches - toothaches [SOME ELEMENTS OMITTED HERE]</p> <p><b>ACTIVE INGREDIENTS</b> Each tablet contains: - 500 mg of aspirin - 32 mg of caffeine</p> <p><b>DIRECTIONS</b> Children above 12 and adults: Take 2 tablets with a full glass of water. This drug can be taken every 6 hours while symptoms persist. Do not exceed 8 tablets in 24 hours. Children under 12: CONSULT A DOCTOR BEFORE GIVING THIS PRODUCT TO CHILDREN UNDER 12.</p> <p><b>POSSIBLE SIDE-EFFECTS</b> Consult a doctor immediately if any of the following side effects occur, as they could be signs of a serious condition: - new symptoms - ringing in the ears - loss of hearing [SOME ELEMENTS OMITTED HERE]</p> <p><b>WARNINGS</b> <b>Drug interactions.</b> A doctor should be consulted before taking this drug if you are already taking a prescription drug for at least one of the following: - anticoagulation - diabetes [SOME ELEMENTS OMITTED HERE] <b>Product warnings. DO NOT TAKE THIS DRUG IF YOU ARE ALLERGIC TO ASPIRIN.</b> This product should not be taken for more than 10 days without consulting a doctor. Consult your doctor if pain persists or gets worse. <b>Particular conditions.</b> A doctor should be consulted before taking this drug if you have any of the following conditions: - asthma - stomach problems [SOME ELEMENTS OMITTED HERE] <b>Children and teenagers.</b> Aspirin should be administered under medical advice only to children and teenagers with symptoms of a virus infection, as it can increase the risks of a serious illness called Reye's Syndrome. <b>Pregnancy.</b> Pregnant women should consult a doctor before taking this drug. Using aspirin during the last 3 months of pregnancy may cause problems to the unborn child or complications during delivery. <b>Alcohol.</b> A doctor should be consulted about the risks associated with alcohol consumption before taking this drug, because aspirin may cause stomach bleeding.</p>
--	--

Figure 4: Anacin patient leaflet: on the left, the online version found on www.drugstore.com; on the right, a normalized version

nal strings.<sup>6</sup> Another mode would be to re-enter the authoring mode of MDA to allow the expert to finish the normalization manually, which would be necessary in the case of incomplete input documents.

## 5.6 Normalization example

Figure 3 shows the abstract tree obtained after normalization that corresponds to the product warnings section of the normalized leaflet on figure 4<sup>7</sup> using a complete grammar that includes the extract on figure 1. Text fragments used as evidence to construct this abstract tree have been highlighted on the input document and connected to their normalized reformulations.

## 6 Discussion and future work

Our research work has still a lot of questions to address, some of which requiring a full implementation of our prototype system. First and foremost, the issue of what allows documents from a given class to be normalized, and what the implications are, have to be more formally defined, before the issues of scalability and portability can be addressed. Then the notion of level of confidence for the automatic analysis has to be defined taking as parameters the class of documents, the grammar used, and the source of the input documents.

The human expert involved in the interactive part guarantees the validity of the whole normalization process. It is in fact an interesting characteristic of our approach, as the result of a normalization can be inspected by comparing two documents as those on figure 4. It is however very important to minimize the time and efforts needed from the expert, so to have the system perform as much filtering as possible. To this end, the reuse of the interactive disambiguation of difficult cases through supervised learning seems particularly important.

**Acknowledgements** The author wishes to thank Marc Dymetman and Christian Boitet for their su-

pervision of his PhD. This work is supported by a grant from ANRT.

## References

- James Allen. 1995. *Natural Language Understanding*. Benjamin/Cummings Publishing, 2nd edition.
- Caroline Brun, Marc Dymetman, and Veronika Lux. 2000. Document Structure and Multilingual Authoring. In *Proceedings of INLG 2000, Mirzpe Ramon, Israel*.
- Marc Dymetman, Veronika Lux, and Aarne Ranta. 2000. XML and Multilingual Document Authoring: Convergent Trends. In *Proceedings of COLING 2000, Saarbrücken, Germany*.
- Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarrán. 1998. Indexing with WordNet Synsets Can Improve Text Retrieval. In *Proceedings of the COLING/ACL Workshop on the Usage of WordNet in Natural Language Processing Systems, Montréal, Canada*.
- Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning. In *Proceedings of EMNLP/VLC'99, College Park, USA*.
- Aurélien Max and Marc Dymetman. 2002. Document Content Analysis through Inverted Generation. In *Proceedings of the workshop on Using (and Acquiring) Linguistic (and World) Knowledge for Information Access of the AAAI Spring Symposium Series, Stanford University, USA*.
- Aurélien Max. 2002. Normalisation de Documents par Analyse du Contenu à l'Aide d'un Modèle Sémantique et d'un Générateur. In *Proceedings of TALN-RECTAL 2002, Nancy, France*.
- Aurélien Max. 2003. Multi-language Machine Translation through Document Normalization. To appear in the proceedings of the EACL'03 EAMT workshop, Budapest, Hungary.
- Andrei Mikheev. 2000. Document centered approach to text normalization. In *Research and Development in Information Retrieval*, pages 136–143.
- G. Miller, R. Berckwith, C. Fellbaum, D. Gross, K. Miller, and R. Teng. 1993. *Five papers on WordNet*. Princeton University Press.
- Nils J. Nilsson. 1998. *Artificial Intelligence: a New Synthesis*. Morgan Kaufmann.
- OVP Editions du VIDAL, editor. 1998. *Le VIDAL de la famille*. Hachette, Paris.
- Daniel S. Paiva. 2000. Investing Style in a Corpus of Pharmaceutical Leaflets: Result of a Factor Analysis. In *Proceedings of the ACL Student Research Workshop, Hong Kong*.
- Fernando Pereira and David Warren. 1980. Definite Clauses for Language Analysis. *Artificial Intelligence*, 13.
- Richard Power and Donia Scott. 1998. Multilingual Authoring using Feedback Texts. In *Proceedings of COLING/ACL-98, Montreal, Canada*.
- Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.
- Ehud Reiter. 1995. NLG Vs Templates. In *Proceedings of ENLGW-95, Leiden, The Netherlands*.

<sup>6</sup>This non-determinism would simply be ignored in the authoring mode, and alternative formulations present in the semantic structure of a candidate content representation would ultimately be changed to their normalized formulation.

<sup>7</sup>Our system being under development, the normalized version of the leaflet shown has been disambiguated manually and is given for illustration purpose only.