# Controlled Natural Language Simplifying Language Use

# Workshop Programme

- Introduction

  09:00 – 09:25 Key-Sun Choi, Hitoshi Isahara
  *"Workshop Introduction"*

- Session 1 : CNL and Controlled Editing

  09:25 – 09:50 Pierrette Bouillon, Liliana Gaspar, Johanna Gerlach, Victoria Porro, Johann Roturier
  *"Pre-editing by Forum Users: a Case Study"*

  09:50 – 10:15 Wai Lok Tam, Yusuke Matsubara, Koiti Hasida, Motoyuki Takaai, Eiji Aramaki, Mai Miyabe, Hiroshi Uozaki
  *"Generating Annotated Corpora with Autocompletion in a Controlled Language Environment"*

- Session 2 : CNL Language Resource and Content Management

  10:15 – 10:30 Delyth Prys, David Chan, Dewi Bryn Jones
  *"What is the Relationship between Controlled Natural Language and Language Registers?"*

- Coffee Break

  10:30 – 11:00

- Invited Talk

  11:00 – 11:30 Teruko Mitamura (CMU, Language Technologies Institute)

- Session 2 Continued : CNL Language Resource and Content Management

  11:30 – 11:55 Kara Warburton
  *"Developing Lexical Resources for Controlled Authoring Purposes"*

  11:55 – 12:20 Giovanni Antico, Valeria Quochi, Monica Monachini, Maurizio Martinelli
  *"Marrying Technical Writing with LRT"*

- Session 3 : ISO Standard for CNL

  12:20 – 13:00 Key-Sun Choi, Hitoshi Isahara
  *"Toward ISO Standard for Controlled Natural Language"*

## Editors

| | |
|---|---|
| Hitoshi Isahara | Toyohashi University of Technology |
| Key-Sun Choi | KAIST |
| Shinhoi Lee | KAIST |
| Sejin Nam | KAIST |

## Workshop Organizers

| | |
|---|---|
| Key-Sun Choi | KAIST |
| Hitoshi Isahara | Toyohashi University of Technology |
| Christian Galinski | Infoterm |
| Laurent Romary | INRIA |

## Workshop Programme Committee

| | |
|---|---|
| Key-Sun Choi | KAIST |
| Hitoshi Isahara | Toyohashi University of Technology |
| Christian Galinski | Infoterm |
| Laurent Romary | INRIA |

# Table of Contents

# Author Index

# Preface

The study of controlled natural language has a long history due to its commercial impact as well as its effectiveness in applications like machine translation, librarianship, information management, terminology management, mobile communication, legal documents, and so on. On the other hand, “text simplification” is also beneficial for efficient communication with respect to all kinds of language use in the Web, such as simplified English Wikipedia for instance. The current progress of linked data also assumes a great potential for knowledge acquisition from text and web data, for example, NLP2RDF and its NIF (http://nlp2rdf.org). It is also obvious that its data fusion and knowledge fusion are more beneficial from the controlled or simplified text or structured source. There is also a working item on this topic in ISO/TC37 “Terminology and other language and content resources” for recommending the principles of controlled natural language and its supporting environment and utilization. This workshop is altogether to know about the scope of controlled natural language for simplifying use in the aspects of their pre-editing for controlled natural language use, their language resources and content management systems in technical writing and mobile life, and the interoperability and relation in the context of standardization. As a result, the workshop may identify the environment of controlled natural language use, their guideline to use, relationship with language resources and other systems, interlinking interoperability and dependency with other standards and activities, and discovery of controlled natural language for human technical writing as well as for the knowledge acquisition and knowledge fusion processes manually and/or automatically by computing and linking in the web environment. It is also observable to see the cooperative work items, to identify the shared tasks to work together open and to analyze their units for simplifying use of language.

# Toward ISO Standard for Controlled Natural Language

**Key-Sun Choi[1], Hitoshi Isahara[2]**
[1]Korea Advanced Institute of Science and Technology
[2]Toyohashi University of Technology
E-mail: kschoi@kaist.edu, isahara@tut.jp

### Abstract

This standard is the first part of the series of ISO standards that are targeted at controlled natural language (CNL) in written languages. It focuses on the basic concepts and general principles of CNL that apply to languages in general. It will cover properties of CNL and CNL classification scheme. The subsequent parts will, however, focus on the issues specific to particular viewpoint and/or applications, such as particular CNLs, CNL interfaces, implementation of CNLs, and evaluation techniques for CNL.

**Keywords:** controlled natural language, ISO standard

## 1. Scope

This standard is the first part of the series of ISO standards that are targeted at controlled natural language (CNL) in written languages. It focuses on the basic concepts and general principles of CNL that apply to languages in general. It will cover properties of CNL and CNL classification scheme. The subsequent parts will, however, focus on the issues specific to particular viewpoint and/or applications, such as particular CNLs, CNL interfaces, implementation of CNLs, and evaluation techniques for CNL.

The objective of this standard is to provide language-independent and general purpose guidelines to enable written texts to be controlled, in a reliable and reproducible manner, in order to suit to specific situation. In language-related activities in industry, the usability of documents is a fundamental and necessary concept. It is thus critical to have a universal definition of what and how one can control language for the purposes of satisfying real world needs. Controlled natural language can be realized by a selection or a simplification of lexicons and/or linguistic rules, or a modification of lexicons and/or linguistic rules. It also can be realized by adding some syntactic or semantic tags to original texts.

There are many applications and fields that need to control language, including machine translation (MT), information retrieval (IR) and technical communication (TC).

## 2. Objectives of controlled natural language

### 2.1 Purposes of controlled natural language

There are several target applications for which CNL will be developed, such as authoring, language learning and man-machine interface or interaction.

### 2.2 Beneficiary of controlled natural language

There are several candidates of beneficiaries of CNL. CNL can be used by human and/or by machine.

1) By human, in more detail;
   Writer, editor, translator and/or reader
   Native speaker or Non-native speaker
   Knowledgeable person or Non-knowledgeable person
   Handicapped people
2) By machine, in more specific;
   Machine Translation, word processor
   Text understanding
   Information Retrieval (text which can be retrieved properly and inquiry which can be converted into proper structure of keywords would be covered by this standardization.)

### 2.3 What is improved by controlled natural language

The general rules and principles of this standard constitute a systematic approach that makes cross-language and cross-domain as well cross-system applications of CNLs more effective.

CNL can be aimed to;
   Improve readability;
   Reduce ambiguity;
   Speed up reading;
   Be easier to comprehend, e.g.,
      Improvement of comprehension for people whose first language is not the language of the document at hand;
      Improvement of comprehension for people with different domain or application background;
      Disambiguation (to what extent and for what purpose);
      Avoiding misunderstanding.
   Reduce cost of whole process of an application, e.g.
      Making human translation and localization easier, faster and more cost effective;
      Being used by computer-assisted translation and machine translation.

### 2.4 Re-usability of Written Text

Another benefit will be re-usability of written text, e.g., re-usability of language resources in larger application scenarios, like Semantic Web or decision-support systems.

Here again, there are several aspects; e.g. documents, texts, sentences, phrases, and terms, which can be easily retrieved and/or modified for re-use.

This aspect is especially useful for CNL in industrial scene.

## 3. Classification of controlled natural language

This standard can be used for making CNL document, changing pre-existing documents to CNL document, and rewriting or re-producing based on the existing text. As for making CNL document, it can be used as a guideline for an author during human writing process, or can be utilized by a system which control language and assist human to write texts.

We can classify "control" by several different viewpoints. The guidelines for CNL will be divided according to the users, such as (1) professional writers, (2) translators, (3) novice (casual users), and (4) machine translation systems.

According to the linguistic structure, there are several levels of CNL, such as (1) syntax, (2) terminology, and (3) document style. More precisely, there are;

Morphological, lexical, syntax, semantics, vocabulary
Character level vs. word level (language specific)
Content Reduction, Sentence Segmentation
Clarification
Style

We can also classify CNL based on its target domain of standards, such as (1) language, (2) user manual, and (3) know-how documents.

Additionally, we should think about "cross" aspects, such as language, domain and system (or application).

CNL can be used as a guideline for generation, such as narrative generation and CNL generated by computer.

It also has educational aspect, i.e. CNL for language learning purpose.

## 4. General principles of controlled natural language

### 4.1 Two viewpoints for CNL

There are at least two kinds of CNL from different viewpoints;

1) Principle for human comprehension
  1-1) Comprehension by human readers
  1-2) Re-usability by human writers
2) Principle for computational viewpoint
  2-1) Machine Translation: Restricted language for MT input
  2-2) CNL for Information Retrieval: simple sentence to be easily retrieved, understandable input queries

### 4.2 Four Sets of Principles of CNL

Four sets of language-independent principles for validating controlled natural language are described in this clause:

1) Universal Principles
  1-1) Cost of whole process of an application
  1-2) Principle of complexity
2) Human-oriented Principles
  2-1) Comprehension
  2-2) Reusability (easy to edit)
3) Computer-oriented Principles
  3-1) Comprehension
  3-2) Reusability (easy to edit)
4) Object-oriented Principles
  4-1) Lexical
  4-2) Sentential

## 5. Related concepts

Simplified language is a language which is generated as a result of some procedure, therefore simplified itself is the objective of this concept, and simplified natural language has no aspect of multilinguality. On the other hand, CNL represents some procedure which restrict some aspects of language phenomena. CNL is not necessarily to be simplified language, and simplified language is a CNL.

# Pre-editing by Forum Users: a Case Study

**Pierrette Bouillon[1], Liliana Gaspar[2], Johanna Gerlach[1], Victoria Porro[1], Johann Roturier[2]**

[1]Université de Genève FTI/TIM - 40
bvd Du Pont-d'Arve, CH-1211 Genève 4, Suisse
{Pierrette.Bouillon, Johanna.Gerlach, Victoria.Porro}@unige.ch

[2]Symantec Ltd.
Ballycoolin Business Park, Blanchardstown, Dublin 15, Ireland
{Liliana_Gaspar, Johann_Roturier}@symantec.com

## Abstract

Previous studies have shown that pre-editing techniques can handle the extreme variability and uneven quality of user-generated content (UGC), improve its machine-translatability and reduce post-editing time. Nevertheless, it seems important to find out whether real users of online communities, which is the real life scenario targeted by the ACCEPT project, are linguistically competent and willing to pre-edit their texts according to specific pre-editing rules. We report the findings from a user study with real French-speaking forum users who were asked to apply pre-editing rules to forum posts using a specific forum plugin. We analyse the interaction of users with pre-editing rules and evaluate the impact of the users' pre-edited versions on translation, as the ultimate goal of the ACCEPT project is to facilitate sharing of knowledge between different language communities.

**Keywords:** pre-editing, statistical machine translation, user-generated content, language communities

## 1. Introduction

Since the emergence of the web 2.0 paradigm, forums, blogs and social networks are increasingly used by online communities to share technical information or to exchange problems and solutions to technical issues. User-generated content (UGC) now represents a large share of the informative content available on the web. However, the uneven quality of this content can hinder both readability and machine-translatability, thus preventing sharing of knowledge between language communities (Jiang et al, 2012; Roturier and Bensadoun, 2011).

The ACCEPT project (http://www.accept-project.eu/) aims at solving this issue by improving Statistical Machine Translation (SMT) of community content through minimally-intrusive pre-editing techniques, SMT improvement methods and post-editing strategies, thus allowing users to post questions or benefit from solutions on forums of other language communities. Within this project, the forums used are those of Symantec, one of the partners in the project. Pre-editing and post-editing are done using the technology of another project partner, the Acrolinx IQ engine (Bredenkamp et al, 2000). This rule-based engine uses a combination of NLP components and enables the development of declarative rules, which are written in a formalism similar to regular expressions, based on the syntactic tagging of the text.

Within the project, we used the Acrolinx engine to develop different types of pre-editing rules for French, specifically designed for the Symantec forums. Primarily, the aim of pre-editing in this context is to obtain a better translation quality in English without retraining the system with new data. In previous work, we have found that the application of these rules significantly improves MT output quality, where improvement was assessed through human comparative evaluation (Gerlach et al, 2013a; Seretan et al, to appear). Another study suggested that for specific phenomena, for example for the register mismatch between community content and training data, pre-editing produces comparable if not better results than retraining with new data (Rayner et al, 2012). Further work (Gerlach et al, 2013b) has shown that pre-editing rules that improve the output quality of SMT also have a positive impact on bilingual post-editing time, reducing it almost by half.

However, it is still unclear whether pre-editing can successfully be implemented in a forum, which is the real life scenario targeted by the ACCEPT project. In the previous studies, the pre-editing rules were applied by native speakers with a translation background, i.e., with excellent language skills. In contrast, in the targeted scenario, the pre-editing task will have to be accomplished by the community members themselves. Although the task was simplified as much as possible for the forum users, by integration of a checking tool in the forum interface, it still involves choosing among one or multiple suggestions, or even correcting the text manually, following instructions when no reliable suggestions can be given. Applying these changes might prove difficult for users with varied linguistic knowledge, as it can involve quite complex modifications, for example restructuring a sentence to avoid a present participle. Another aspect to consider is the motivation of the users: if pre-editing requires too much time or effort, users will be less inclined to complete this step. Additionally, as users probably have little knowledge of the functioning of an SMT engine or the consequences of pre-editing, the importance of making certain changes to the source will not be obvious to them.

The aim of this study is therefore to ascertain whether light pre-editing rules which were developed using the Acrolinx formalism and which have proved to be useful for SMT can be applied successfully by forum users.

In the rest of the paper, Section 2 provides more details about the French Acrolinx pre-editing rules developed for the Symantec forums. Section 3 describes the experimental setup and provides details about the experiments conducted for evaluating the rules with forum users. In Section 4, we discuss the results obtained in these experiments and, finally, conclusions and directions for future work are provided in Section 5.

## 2. Pre-editing in ACCEPT

Pre-editing can take different forms: spelling and grammar checking; lexical normalisation (e.g. Han & Baldwin, 2011, Banerjee et al., 2012); Controlled Natural Language (CNL) (O'Brien, 2003; Kuhn, 2013); or reordering (e.g. Wang et al, 2007; Genzel, 2010). However, few pre-editing scenarios combine these different approaches. For partially historical reasons, CNL was mostly associated with rule based machine translation (RBMT) (Pym, 1988; Bernth & Gdaniec, 2002; O'Brien & Roturier, 2007; Temnikova, 2011, etc. (one exception is (Aikawa et al, 2007)). On the contrary, spellchecking, normalisation and reordering were frequently used as pre-processing steps for SMT. In this work, the particularities of community content have led us to choose an eclectic approach. We developed rules of all the types mentioned above which answer the following criteria:

- The rules focus on specificities of community content that hinder SMT, namely informal and familiar style (not well covered by available training data), word confusion (related to homophones) and divergences between French and English.
- As we cannot reasonably ask forum users, whose main objective is obtaining or providing solutions to technical issues, to painstakingly study pre-editing guidelines, compliance with the rules must be checked automatically. Therefore rules must be implemented within a checking tool, in our case Acrolinx. This entails some restrictions, especially due to the nature of the Acrolinx formalism, which is for example not well suited to detect non local phenomena. On the positive side, it also means that rules are easily portable to other similar tools since they don't require a lot of linguistic resources.
- Another condition for successful rule application by forum users is that suggestions are provided, since we cannot expect forum users to reformulate based only on linguistic instructions (such as "avoid the present participle", "avoid direct questions", "avoid long sentences", etc). For this reason, common CNL rules like "avoid long sentences" were replaced by more specific rules, accompanied by an explanation which appears on a tooltip. A good example is the rule which replaces ", ce qui", by a full stop followed by a pronoun: ". Ceci" (see **Figure 1**).

N360 sauvegarde les fichiers en plusieurs répertoires, ce qui peut parait abscons, mais c'est correct.
N360 sauvegarde les fichiers en plusieurs répertoires. Ceci peut paraître abscons, mais c'est correct.

**Figure 1**. Example of pre-editing rule used to substitute traditional CNL rules like "avoid long sentences"

In the absence of forum post-edited data that would have allowed identification of badly translated phrases or phenomena, the rules were developed mainly using a corpus-oriented approach. Two specific resources proved to be particularly useful: the out-of-vocabulary (OOV) items, which are a good indicator of the data that is not covered in the training set (see Banerjee et al, 2012), and the list of frequent trigrams and bigrams, present in the development data but absent from the training corpus.

Three sets of rules were developed intended to be used in sequence. A first distinction is made between rules for humans (which also improve source quality) and rules for the machine (which can degrade it or change it considerably since the only aim is to improve MT output) (Hujisen, 1998). The rules for humans were split up into two sets, according to the pre-editing effort they require.

A first set (Set1) contains rules that can be applied automatically. This set includes rules that treat unambiguous cases and have unique suggestions. It contains rules for homophones, word confusion, tense confusion, elision and punctuation. While the precision of the rules included in this set is reasonably high, it is not perfect. The automatic application of this set does therefore produce some errors that might be avoided if the rules were applied manually instead. Examples of rules contained in this set are given in **Table 1**.

| **Rule** | **Raw** | **Pre-edited** |
|---|---|---|
| *Confusion of the homophones "sa" and "ça"* | oups j'ai oublié, j'ai **sa** aussi. | oups j'ai oublié, j'ai **ça** aussi. |
| *Missing or incorrect elision* | Lancez Liveupdate et regardez **si il** y a un code d'erreur. | Lancez Liveupdate et regardez **s'il** y a un code d'erreur. |
| *Missing hyphenation* | Il est **peut être** infecté, ce qui serait bien dommage. | Il est **peut-être** infecté, ce qui serait bien dommage. |

**Table 1.** Examples for Set1

A second set (Set2) contains rules that have to be applied manually as they have either multiple suggestions or no suggestions at all. The rules correct agreement (subject-verb, noun phrase, verb form) and style (cleft sentences, direct questions, use of present participle, incomplete negation, abbreviations), mainly related to informal/familiar language. The human intervention required to apply these rules can vary from a simple selection between two suggestions, to manual changes, for example for checking a bad sequence of words. Examples of rules contained in this set are given in **Table 2**.

| Rule | Raw | Pre-edited |
|---|---|---|
| *Avoid direct questions*<br>*Avoid abbreviations* | **Tu as** lu le **tuto** sur le forum? | **As-tu** lu le **tutoriel** sur le forum? |
| *Avoid the present participle* | Certains jeux **utilisant** Internet ne fonctionnent plus. | Certains jeux **qui utilisent** Internet ne fonctionnent plus. |
| *Avoid letters between brackets* | Regarde le**(s)** barre**(s)** que tu as téléchargées et surtout le**(s)** site**(s)** web où tu les as récupérés. | Regarde les barres que tu as téléchargées et surtout les sites web où tu les as récupérés. |

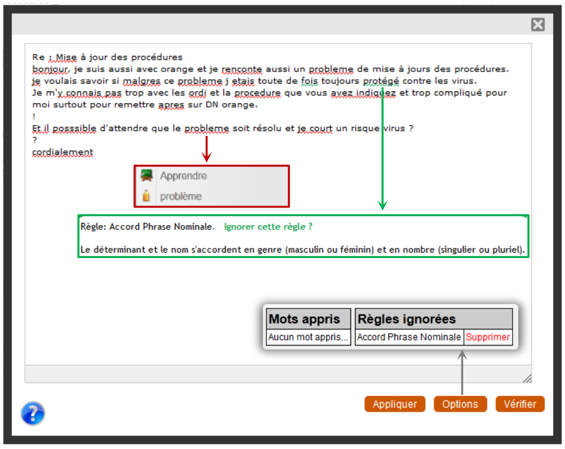
**Table 2**. Examples for Set2

Finally, the rules for the machine were grouped in a third set (Set3) that is applied automatically and will not be visible to end-users. These rules modify word order and frequent badly translated words or expressions to produce variants better suited to SMT. The rules developed in this framework are specific to the French-English combination and to the technical forum domain. Examples of rules contained in this set are given in **Table 3**.

| Rule | Raw | Pre-edited |
|---|---|---|
| *Avoid informal 2$^{nd}$ person* | J'ai apporté une modification dans le titre de **ton** sujet. | J'ai apporté une modification dans le titre de **votre** sujet |
| *Replace pronoun by "ça"* | Il est recommandé de **la** tester sur une machine dédiée. | Il est recommandé de tester **ça** sur une machine dédiée. |
| *Avoid "merci de"* | **Merci de** nous tenir au courant. | **Veuillez** nous tenir au courant. |

**Table 3**. Examples for Set3

In ACCEPT, pre-editing is completed through the ACCEPT plugin directly in the Symantec forum. This plugin was developed using Acrolinx's technologies and specifically conceived to check the compliance with the rules directly where content is created (ACCEPT Deliverable D5.2, 2013). This plugin "flags" potential errors or structures by underlining them in the text. Depending on the rules, when hovering with the mouse cursor over the underlined words or phrases, the user receives different feedback to help him apply the rule correction (**Figure 2**). For rules with suggestions, a contextual menu provides a list of potential replacements, which can be accepted with a mouse click. For rules without suggestions, a tool-tip comes up with the description of the error but no list of potential replacement is provided. Modifications then have to be done directly by editing the text. Besides these two main interactions, users can also choose to "learn words", i.e. add a given token to the system so that it will not be flagged again, or "ignore rules", i.e. completely deactivate a given rule. Both actions are stored within the user profile and remain active for all subsequent checking sessions. By means of a properties window, users can view learned words and ignored rules, which can be reverted at any time. **Figure 2** shows the plugin in action.



Re : Mise à jour des procédures
bonjour, je suis aussi avec orange et je renconte aussi un probleme de mise à jours des procédures.
je voulais savoir si malgres ce probleme j etais toute de fois toujours protégé contre les virus.
Je m'y connais pas trop avec les ordi et la procedure que vous avez indiquez et trop compliqué pour moi surtout pour remettre apres sur DN orange.
!
Et il possible d'attendre que le probleme soit résolu et je court un risque virus ?
?
cordialement

Règle: Accord Phrase Nominale. Ignorer cette règle ?

Le déterminant et le nom s'accordent en genre (masculin ou féminin) et en nombre (singulier ou pluriel).

| Mots appris | Règles ignorées | |
|---|---|---|
| Aucun mot appris... | Accord Phrase Nominale | Supprimer |

**Figure 2**. ACCEPT pre-editing plugin used for this study

In this study, our aim is twofold. In a first step, we want to compare rule application by forum users and experts. In a second step, we wish to determine if it is preferable to have a semi-automatic, yet not entirely reliable process (where Set1 is applied automatically), or a manual process where all the rules from Set1 and Set2 are checked manually. This last approach will strongly depend on the motivation and skills of the users. These different scenarios (user vs expert, manual vs automatic) will be compared in terms of pre-editing activity (number of changes made in the source and the target) and in terms of the impact of changes on translation output. This impact will be evaluated using human comparative evaluation. In the next section, we will describe the experimental setup for the scenarios mentioned above.

## 3. Experimental Setup

### 3.1 Pre-editing

In order to compare the different pre-editing scenarios, we collected the following pre-edited versions of our corpus:

**UserSemiAuto**: Rules from Set1 were applied automatically. Then, the corpus was submitted to the forum users, who applied the rules from Set2 manually using the ACCEPT plugin.

**UserAllManual**: The raw corpus was submitted to the forum users, who applied the rules from Set1 and Set2 manually using the ACCEPT plugin. This version was produced at one week interval from UserSemiAuto.

**Expert**: Rules from Set1 were applied automatically. Then, the corpus was submitted to a native French speaking language professional, who applied the rules from Set2 manually.

**Oracle**: This version is the result of manual post-processing of the Expert version by a native French speaker. All remaining grammar, punctuation and spelling issues were corrected. No style improvements were made in this step.

For the User scenarios, the pre-editing activity was recorded using the ACCEPT plugin. This included recording the number and type of errors flagged by the rules and the actions performed during the process (accepted suggestions, displayed tooltips, ignored rules and words learned). The output data was collected in a JSON format.

To complete the pre-editing process as designed for ACCEPT, once all manual pre-editing steps were performed, we applied the rules from Set3 automatically to all pre-edited versions. All versions were then translated into English using the project's baseline system, a phrase-based Moses system, trained on translation memory data supplied by Symantec, Europarl and news-commentary (ACCEPT Deliverable D4.1, 2012). We then set up five human comparative evaluations on Amazon Mechanical Turk and measured the pre-editing activity as explained in the following section.

## 3.2 Evaluation

### 3.2.1 MT output

For the comparative evaluations, the test data was split into sentences. We presented three bilingual judges with sentence pairs in randomised order. These sentences are translations of different pre-edited versions of the same source sentence. Sentences with identical translations were not included in the evaluation. The judges were asked to assign a judgement to each pair on a five-point scale {first clearly better, first slightly better, about equal, second slightly better, second clearly better}. The majority judgement for each sentence was calculated.

The evaluations were performed on Amazon Mechanical Turk, using the same setup as in previous studies (Rayner et al, 2012; Gerlach et al, 2013a). Tasks were restricted to workers residing in Canada and having a reliable work history on AMT. We chose to use AMT workers for this evaluation because we have found that for simple tasks like these, the results obtained are reliable and can be obtained fast.

We first compared the translations of the Raw with the translations of the version pre-edited by the Expert (**Raw vs Expert**). The result was used as a baseline for the evaluations of the User versions and allowed us to corroborate the positive impact of our rules on translation and validate the results obtained in previous studies (Gerlach et al, 2013a, 2013b).

In a second evaluation, we compared the translations of the different User versions with the translations of the Raw (**Raw vs User**), to evaluate the impact of our rules when applied by users.

In a third evaluation, we compared the translations of the different User versions against the Expert version (**Users vs Expert**), in order to complement results obtained with the second evaluation.

A fourth evaluation was designed to determine the impact of applying some of the rules automatically, as opposed to performing an entirely manual application. For this evaluation, we asked judges to compare the translations produced in each scenario, UserSemiAuto and UserAllManual, for a same user (**UserSemiAuto vs UserAllManual**).

Finally, we compared the translations of the Raw with the translations of the Oracle version (**Raw vs Oracle**). This allowed us to assess the potential of correcting all grammar, punctuation and spelling issues that are not covered by our rules.

### 3.2.2 Pre-editing activity

In order to gain more insight into the effort required for applying pre-editing rules, we performed a quantitative analysis of the activities logged by the plugin during the pre-editing process. We looked at the number of flagged errors (errors found) and the total number of actions performed by users. We also investigated the acceptance rate of suggestions as well as the rules and words which had been ignored and/or learned. Additionally, we calculated the Levenshtein distance between the raw and the pre-edited User versions to quantify the total tokens changed during pre-editing. We compared results per scenario and per user.

## 3.3 Data selection

The amount of data we could reasonably expect volunteer forum users to process being limited, we chose to create a corpus of about 2500 words for this study. From an initial corpus of 10000 forums posts,  only posts of 250 words or less were selected to ensure that the final corpus would contain posts with a diversity of writers and topics. Among these, we then chose to select posts with a relatively high occurrence of errors and structures to pre-edit. Focussing on posts with many errors allowed us to cover a larger number of pre-editing rules, and thereby increase the chances that users would treat or reflect upon a diversity of rules, giving us more insight into the difficulties encountered with each rule category. To this end, we processed our corpus with the Acrolinx Batch Checker, which produces reports that summarise all the errors found for each rule. In Acrolinx, rules are grouped in three categories: grammar, style and spelling. For this study, we chose to focus on grammar and style rules, as the application of these is more likely to cause difficulties to our participants, as opposed to spelling, which works like any other spelling checker that most users are familiar with. Therefore, we kept only posts with at least 3 grammar and 3 style errors (mean number of errors per post: 5.7). Among these, we selected the posts with the highest error/words ratio, resulting in a set of 25 posts. These posts were made available to users of the French Norton forum[1] in the forum itself to maximize the ecological validity of the study. Specific forum sections were created for each participant and automatically populated with the selected posts using the Lithium API.[2] In this study users were asked to edit texts that they had not necessarily authored,

[1] http://fr.community.norton.com

[2] http://www.lithium.com/products/technology/integration

which would not be the case in a real-life scenario.

### 3.4 User selection

To recruit users willing to participate in our study, we made an open call for participation in the French-Speaking Norton forum. We did not look for any specific profile. The only prerequisite was to be a French native speaker. 7 users showed their willingness to participate and were contacted, but only 2 had completed all tasks at the time of this study.

## 4. Results

In this section we present the results of the evaluations for the two main research questions (Users vs Expert and SemiAuto vs AllManual) we seek to answer both in terms of translation quality and pre-editing activity.

### 4.1 Users vs Expert

#### 4.1.1 Translation quality

The results obtained for the Expert version through a comparative evaluation confirm those of previous studies, namely that correct application of the pre-editing rules has a significant positive impact on translation quality. **Table 4** shows that for 52% of sentences, the translation of the pre-edited version is better, while the translation is degraded for only 6% of sentences. A McNemar test showed that the difference of cases in which pre-editing had a positive vs a negative impact is statistically significant (p<0.001).

| | identical | raw better | about the same | pre-edited better | no majority judgement |
|---|---|---|---|---|---|
| **Expert** | 32% | 6% | 4% | 52% | 5% |
| **Oracle** | 29% | 6% | 2% | 60% | 3% |

**Table 4**. Raw against Expert pre-edited and Oracle

The Oracle version only produces slightly better results (60%) than the Expert version. This suggests that our light pre-editing rules, in their current state, can produce high-quality results not far from those obtained with the Oracle.

**Table 5** presents the results for the User scenarios. We observe that they are very close to those obtained with Expert pre-editing.

| | identical | raw better | about equal | user better | no majority judgement |
|---|---|---|---|---|---|
| **SemiAuto** | | | | | |
| user1 | 42% | 7% | 2% | 45% | 4% |
| user2 | 41% | 4% | 1% | 50% | 3% |
| **AllManual** | | | | | |
| user1 | 43% | 6% | 2% | 47% | 3% |
| user2 | 44% | 2% | 2% | 50% | 2% |

**Table 5**. Raw against User pre-edited

For both scenarios and users, the translations of nearly half of the sentences are improved by pre-editing. As in the case of the Expert, the difference between improved and degraded sentences is statistically significant (p<0.001).

However, while the number of improved sentences is similar, these results do not tell us if pre-editing by the users produced as good a result as pre-editing by the Expert. It cannot be excluded that, while they were judged as better than the Raw version, some of the improved sentences are still of lesser quality than the Expert version. For this reason, we decided to compare the User versions against the Expert version. Results are shown in **Table 6**.

| | identical | user better | about equal | expert better | no majority judgement |
|---|---|---|---|---|---|
| **SemiAuto** | | | | | |
| user1 | 65% | 5% | 2% | 25% | 3% |
| user2 | 60% | 13% | 4% | 19% | 3% |
| **AllManual** | | | | | |
| user1 | 65% | 10% | 3% | 19% | 3% |
| user2 | 57% | 12% | 4% | 24% | 3% |

**Table 6**. User against Expert

In all scenarios, flag application performed by the users and the Expert produced identical translations for more than half of the sentences (65%-60%/65%-57%). In all scenarios, the Expert version is considered better than the Users version in less than a quarter of the sentences (19% to 25%). In some cases, the User version is considered better than the Expert. Globally, in three out of four cases the differences are statistically significant (p<0.0001) but small, which suggests that users are not far from the Expert.

#### 4.1.2 Pre-editing activity

In terms of activity performed, the users and the Expert are also close. The comparison of the Levenshtein distance for all versions against Raw (2274 original tokens) shows that users made less changes than the Expert in both scenarios, but again the difference is small. In average, the Expert changed 5% more tokens than the users. This may also be due to the incomplete application of rules. The additional changes made in the Oracle version amount only to 5%. **Table 7** displays the Levenshtein distance from Raw for all scenarios.

| | **User SemiAuto** | **User AllManual** | **Expert** | **Oracle** |
|---|---|---|---|---|
| **Tokens** | 449 (user1) | 465 (user1) | 582 | 694 |
| | 527 (user2) | 480 (user2) | | |
| **% of total** | 20% (user1) | 20% (user1) | 26% | 31% |
| | 23% (user2) | 21% (user2) | | |

**Table 7**. Levenshtein distance from Raw - All scenarios

From Section 4.1 we can then conclude that both users and experts can reach a good pre-editing performance, with a significant impact on SMT.

## 4.2 UserSemiAuto vs UserAllManual

### 4.2.1 Translation quality

For each user, version for scenario 1 (SemiAuto) was compared with version for scenario 2 (AllManual).

| | identical | semi-auto better | about equal | all manual better | no majority judgement |
|---|---|---|---|---|---|
| **user1** | 72% | 8% | 6% | 13% | 0% |
| **user2** | 58% | 18% | 6% | 16% | 2% |

**Table 8**. UserSemiAuto against UserAllManual

**Table 8** shows that for more than half of the sentences, there is no difference between the two versions. The difference between UserSemiAutoBetter and UserAllManualBetter is relatively small and is not statistically significant (McNemar test, p>0.05).

### 4.2.2 Pre-editing activity

The data logged using the ACCEPT plugin provided information about number of flags and actions performed to correct the text in both User scenarios (UserSemiAuto vs UserAllManual).

As expected, users had to deal with more flags in the UserAllManual scenario than in the UserSemiAuto because they had to apply both sets (1 and 2) manually (430 vs 642). This fact required more attention from users, as evidenced by the higher number of actions performed in the UserAllManual scenario (347 and 327 in UserSemiAuto vs 501 vs 512 in UserAllManual). A summary of actions and flags is provided in **Table 9**.

| | **UserSemiAuto** | | **UserAllManual** | |
|---|---|---|---|---|
| | **user1** | **user2** | **user1** | **user2** |
| totalFlags | 430 | | 642 | |
| total actions performed | 347 | 327 | 501 | 512 |
| of which accepted suggestions (%) | 213 (61%) | 211 (65%) | 431 (86%) | 375 (73%) |
| total available suggestions | 333 | | 539 | |
| % of accepted suggestions over total available | 64% | 63% | 80% | 70% |

**Table 9**. Flags and actions logged by the ACCEPT plugin

In both scenarios, suggestions are among the most frequent type of performed actions. They represent 61%-86% of actions for user1 and 65%-73% of actions for user2 (UserSemiAuto and UserAllManual respectively). Moreover, suggestions have a high acceptance rate for both users in both scenarios (64%-80% for user1 and 63%-70% for user2 over the total available suggestions), which suggests that the suggestions provided are considered useful.

The Levenshtein distance for the two user scenarios (UserSemiAuto and UserAllManual) revealed information about the number of edits performed by users in each scenario (see **Table 10** below). In the UserSemiAuto scenario, 141 tokens were changed after the automatic application of Set1 to the raw original corpus. This scenario then required 326 more changes from user1 when applying Set2 manually, and 407 from user2. Conversely, more tokens were changed when applying both Set1 and Set2 manually in the UserAllManual scenario, which shows that more edit activity was required in this scenario: 465 tokens were changed by user1 (+ 39%) and 480 by user2 (+ 17%).

| **Scenario** | | | **Changed tokens** |
|---|---|---|---|
| **Auto application of Set1 to Raw** | | | 141 |
| **User SemiAuto** | manual set2 | **user1** | 326 |
| | | **user2** | 407 |
| **User AllManual** | manual set1&set2 | **user1** | 465 |
| | | **user2** | 480 |

**Table 10**. Levenshtein distance - User scenarios

The conclusion from Section 4.2 is therefore that the high-precision (yet not perfect) rules from Set1 can be safely automatically applied with less effort from users.

## 4.3 Learned words and ignored rules

Considering that we had only two participants and a relatively small amount of data, results presented in this section are too scarce to perform a significant quantitative analysis, but they still provide insights into user preferences. As we suspect that the distinction between "learn word" and "ignore rule" might not have been entirely clear for the users, we have chosen to regroup both cases. In the following, we will call these "rejected flags".

In both scenarios, both users chose to reject a certain number of flags, as shown in **Table 11**.

| | **semiAuto** | **allManual** |
|---|---|---|
| **user1** | 6 | 22 |
| **user2** | 22 | 21 |

**Table 11**. Rejected flags per user

A closer investigation shows that by far the most frequently rejected are spelling flags (14, counted over both users and both scenarios). Among these, only 5 are "real" spelling issues such as missing accents or typos, while the others are either proper nouns, anglicisms or abbreviations, all very common on a technical forum, and not always incorrect. Three of these flags were also rejected by the Expert. Unsurprisingly, the next rule that was rejected frequently is "avoid anglicisms" (13 flags, counted over both users and both scenarios). Words such as "boot", "Trojan" or "software" are very common in French techie speak, and users might not see the use of replacing them with less common French equivalents. The remaining ignored flags are mostly style rules, such as "avoid conjunctions at Beginning of Sentence" and "avoid present participle".

We also examined the impact of flag rejection on translation. However, due to the experimental setup it is not possible to draw direct conclusions, as the evaluation is sentence-based and most of the sentences had several flags. It is therefore not possible to determine whether omission of one flag was the determining change that influenced the evaluation of an entire sentence. We did however find that for 17% of sentences where a flag was rejected, the translation was identical to that obtained with the Expert version where the flags had effectively been applied. It must be noted that in 6 cases, users corrected the flagged word or phrase, despite choosing to ignore the rule or learn the word. This might be due to manipulation errors.

## 5. Conclusion

In this paper, we ascertained that pre-editing rules developed with a light formalism (regular expressions) are sufficient to produce significant improvement on SMT and can be applied successfully by some forum users. In particular, we have found that:

- The two users who participated in this study are close to experts in terms of pre-editing activity and produce significant impact on SMT.

- The semi-automatic process can be safely applied without degrading the quality of the results. Besides, it saves time and effort from users, as less edits and actions are required when Set1 is applied automatically.

- The analysis of interaction with rules allowed us to discriminate between rules that users might be willing to apply from those rules perceived as incorrect or purely stylistic, and thus not essential and time-consuming. This can help in the future to filter out unnecessary rules or to decide which rules to place in an automatic set (a decision which implies increasing precision in detriment of coverage). For example, some rules rejected by users but with a high impact on SMT, as "avoid present participle" could be restricted to be automatic. Further research will be needed in this sense.

## 6. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 288769.

## 7. References

ACCEPT Deliverable D4.1 (2012), http://www.accept.unige.ch/Products/

ACCEPT Deliverable D5.2 (2013), http://www.accept.unige.ch/Products/

Aikawa, T., Schwartz, L., King, R., Corston-Oliver, M., and Lozano, C. (2007). Impact of controlled language on translation quality and post-editing in a statistical machine translation environment. In *Proceedings of the MT Summit XI*, Copenhagen, Denmark, pp. 10-14,

Banerjee, P., Naskar, S. K., Roturier, J., Way, A. and Van Genabith, J. (2012). Domain Adaptation in SMT of User-Generated Forum Content Guided by OOV Word Reduction: Normalization and/or Supplementary Data? In *Proceedings of EAMT*, Trento.

Bernth, A. and Gdaniec, C. (2002). MTranslatability. In *Machine Translation 16*, pp. 175-218.

Bredenkamp, A., Crysmann B., and Petrea, M. (2000). Looking for errors: A declarative formalism for resource-adaptive language checking. In *Proceedings of LREC 2000*. Athens, Greece.

Genzel, D. (2010). Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, China.

Gerlach, J., Porro, V., Bouillon, P., and Lehmann, S. (2013a). La préédition avec des règles peu coûteuses, utile pour la TA statistique des forums ? *In Proceedings of TALN/RECITAL 2013*. Sables d'Olonne, France.

Gerlach, J., Porro, V., Bouillon, P., and Lehmann, S. (2013b). Combining pre-editing and post-editing to improve SMT of user-generated content. In *Proceedings of MT Summit XIV Workshop on Post-editing Technology and Practice*. Nice, France.

Han, B. and Baldwin, T. (2011). Lexical normalisation of short text messages: Makn sens a #twitter, In *ACL 2011*, Portland, OR, USA, pp. 368–378.

Hujisen, W. O. (1998). Controlled Language: An introduction. In *Proceedings of CLAW 98,* Pittsburg, Pennsylvania, pp. 1–15.

Jiang, J., Way, A., and Haque, R. (2012). Translating User-Generated Content in the Social Networking Space. In *Proceedings of AMTA 2012*, San Diego, CA, United States.

Kuhn, T. (2013) A survey and classification of controlled natural languages. *Computational Linguistics*. Early Access publication: June 26, 2013. doi: 10.1162/COLI_a_00168.

O'Brien, S. (2003). Controlling controlled English: An Analysis of Several Controlled Language Rule Sets. In *EAMT-CLAW-03*, Dublin, pp. 105-114.

O'Brien, S. and Roturier, J. (2007). How Portable are Controlled Languages Rules? A Comparison of Two Empirical MT Studies. In *MT Summit XI*, Copenhagen, Denmark, pp. 105-114.

Pym, P. J. (1988). Pre-editing and the use of simplified writing for MT: an engineer's experience of operating an MT system. In *Translating and the Computer 10*.

Rayner, M., Bouillon P. and Haddow B. (2012). Using Source-Language Transformations to Address Register Mismatches in SMT. In *Proceedings of AMTA*, San Diego, CA, United States.

Roturier, J., and Bensadoun, A. (2011). Evaluation of MT Systems to Translate User Generated Content. In *Proceedings of the MT Summit XIII*, p. 244-251.

Roturier, J., Mitchell, L., and Silva, D. (2013). The ACCEPT Post-Editing Environment: a Flexible and Customisable Online Tool to Perform and Analyse Machine Translation Post-Editing. In *Proceedings of MT Summit XIV Workshop on Post-Editing Technology and Practice*. Nice, France.

Ruffino, J.R. (1982). Coping with machine translation. In:

Veronica Lawson (ed.) *Practical Experience of Machine Translation*: *Proceedings of a Conference*, pp. 57-60.

Seretan, V., Bouillon P. and Gerlach J. (to appear). A Large-Scale Evaluation of Pre-editing Strategies for Improving User-Generated Content Translation. In *LREC 2014*.

Streiff, A. A. (1985). New developments in TITUS 4. In: Veronica Lawson (ed.) *Tools for the Trade: Translation and the Computer Aslib*, London, United Kingdom, pp. 185-192.

Temnikova, I (2011). Establishing Implementation Priorities in Aiding Writers of Controlled Crisis Management Texts. In *Recent Advances in Natural Language Processing (RANLP 2011),* Hissar, Bulgaria, pp. 654-659.

Wang, C., Collins, M. and Koehn, P. (2007). Chinese Syntactic Reordering for Statistical Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL), pp. 737-745.

# Generating Annotated Corpora with Autocompletion in a Controlled Language Environment

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

[1]The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan

[2]Fuji Xerox Company Limited
9-7-3 Akasaka, Minato-ku, Tokyo, Japan

[3]Kyoto University
Yoshidahonmachi, Sakyo-ku, Kyoto, Japan

[4]Teikyo University
2-11-1 Kaga, Itabashi-ku, Tokyo, Japan

{tam, matsubara}@sict.i.u-tokyo.ac.jp, hasida.koiti@i.u-tokyo.ac.jp
Motoyuki.Takaai@fujixerox.co.jp, {aramaki,miyabe}@yoslab.net, uozaki@m.u-tokyo.ac.jp

## Abstract

This paper presents a novel attempt to generate annotated corpora by making use of grammar based autocompletion. The idea is to automatically generate corpora on the fly while a user is working on his own stuff. In the medical domain, this user would be a physician. While he uses an authoring tool to write a pathology report or enters text in an Electronic Healthcare Record (EHR) system, grammar and ontology-based autocompletion is built into such systems such that user input is limited to text parseable by the grammar used for autocompletion. As soon as the user finishes his work, the grammar used for autocompletion would be used for assigning syntactic structures and semantic representations to his input automatically. This way corpora can be generated by limiting annotation and grammar writing by a linguist to help building grammar and ontology-based autocompletion into a EHR system or an authoring tool of pathology reports. After autocompletion starts working hand-in-hand with these applications, new input from users does not need further annotation by human. Users are not supposed to be paid for using an EHR system or an authoring tool with built-in autocompletion that helps them to do their job.

**Keywords:** autocompletion, generation, corpora, grammar

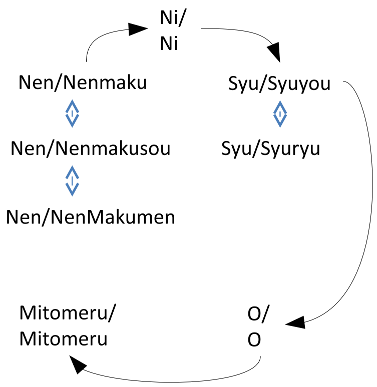


Figure 1: Generation by autocompletion

## 1. Introduction

This paper presents our attempt to kill two birds with one stone by generating annotated corpora with autocompletion. Annotated corpora are useful resources for a lot of subfields of NLP. But creating one such corpora is expensive. It takes times. It is monotonous when all needs to be done is just following some existing annotation guidelines. But it needs an expert to understand the annotation guidelines and tell when the existing annotation guidelines do not cover the text he is annotating. It is true that putting some labels useful for shallow processing can be done by housewives or undergraduates with little training. But anything beyond phrase structures would need post-docs or at least late stage graduate students with proper training. The monotonicity and the expertise it requires cast doubt on the feasibility of proposals for turning annotation into games and doing away with the need to pay annotators at all. One such proposal is given by Chamberlain et al. (2008). It describes a game with the goal of labelling the antecedent of an anaphoric expression. Whether the game is fit to be called a game is the first question we have. Typical computer games and video games come with stories that offer gratification in terms of satisfying experience of the real world like conquering the world or saving the princess. Gambling games give promises of material rewards. But the annotation game described by Chamberlain et al. (2008) does not offer either. More than that, it asks its players to do something that does not require the level of expertise like assigning a phrase strucuture to a sentence or semantic representation to every node of it. So speaking of a game as the answer to the cost issues of all sorts of annotation is an overgeneralization. A less extreme cost-cutting measure than such game is keeping annotation as paid work but switching from skilled labour to unskilled labour (Snow et al., 2008). Still, unskilled labour are given relatively simple tasks that give them very limited choices (less than 10). Unlike these tasks, determining the syntactic structure and semantic representation of a sentence is far more complicated. Ignoring the unlimited possibilities of coming up with new predicates as a result of unrestricted semantic finegrainedness, we still get more than $n!$ possible ways to assign a phrase structure tree to a string of length $n$ (Karl, 1999). For these complicated and expensive tasks, we are not aware of any proposal of cost-cutting methods.

Getting back to our method of reducing the cost of these complicated tasks introduced in the very first sentence, we would like to give a general idea of it here. The idea is to automatically generate corpora on the fly while a user is working on his own stuff. In the medical domain, this user would be a physician. While he uses an authoring tool to write a pathology report or enters text in an Electronic Healthcare Record (EHR) system, grammar and ontology-based autocompletion is built into such systems such that user input is limited to text parseable by the grammar used for autocompletion. As soon as the user finishes his work, the grammar used for autocompletion would be used for assigning syntactic structures and semantic representations to his input automatically. This way corpora can be generated by limiting annotation and grammar writing by a linguist to help building grammar and ontology-based autocompletion into a EHR system or an authoring tool of pathology reports. After autocompletion starts working hand-in-hand with these applications, new input from users does not need further annotation by human. Users are not supposed to be paid for using an EHR system or an authoring tool with built-in autocompletion that helps them to do their job. Based on an authoriing tool of Japanese pathology reports of gastric carcinoma with built-in autocompletion, we give an outline of the whole process in this paper.

## 2. Staring with Autocompletion

This first step is to get autocompletion to start working. Unlike autocompletion built in search engines we are familiar with, the kind of autocompletion we need is built with the goal of controlling generation and completing a grammatical sentence. This casts doubt on the appropriateness of the nobrainer bigram-based approach adopted by Nandi and Jagadish (2007) and followed by many unpublished works. The bigram-based approach guarantees that something will be suggested for all seen words. But it cannot guarantee any sequence of characters to be a sentence as a sequence of the most frequent bigrams can be obtained from different sentences. When user input is not supposed to be a grammatical sentence, as in the case of search engines, there is little point in making sure that user input would eventually reveal itself to be a sentence. When user input is expected to be a sentence, as in the case of our pathology report authoring tool, ignoring the importance of competing a grammatical sentence in pursuit of robustness becomes questionable. The likely result of robust autocompletion, presenting a long list containing none of the words a user wishes to enter, is found to be much more annoying than failing to give any suggestion of the next word to enter. If we go further to give autocompletion a second purpose of generating a corpus, it seems clear that the importance of completing a grammatical sentence should be valued much higher than robustness. So we choose to bind user input to predictions by grammar rules, an idea not very popular but not completely new (Wood, 1996). Our grammar-based approach to autocompletion is found on the left-corner parsing algorithm (Nederhoff, 1993), with some modification:

```
1. Initialize chart variables: IA, AA, PREDS as empty arrays
   IA[POSITION ] inactive arcs starting at POSITION
   AA[POSITION ] active arcs starting at POSITION
   PREDS[POSITION ] list of PRED_ITEM, each of which is a tuple (LEFT_CORNER_LABEL, RIGHT_CORNER_LABEL)), starting at POSITION
       RIGHT_CORNER_LABEL previous label
       LEFT_CORNER_LABEL predicted label
2. assign (ROOT, NONE) to PREDS[0]
3. For POSITION in (0...N)
   (a) For IA_ITEM IN IA[POSITION]:
       i. Create a new arc by applying a grammar rule to IA_ITEM
       ii. If the new arc is active, then add its label to PREDS
       iii. Pack redundant arcs
   (b) FOr AA_ITEM in AA[POSITION]:
        i. Create a new arc by combining an inactive arc with AA_ITEM
        ii. If the new arc is active, then adds its label to PREDS
        iii. Pack redundant arcs
    (c) Search DICTIONARY for the current word and create an inactive arc
4. Extract PRED_ITEM from PREDS[LAST_POSITION]
5. Rank PRED_ITEM in terms of bigrams
```

The grammar on which autocompletion bases is the result of extracting all combinations of categories that form a complete sentence in an annotated corpora. In this corpus, there are 100 pathology reports which can be broken down into 2000 sentences. When annotating this corpus, we do not try any trick that reduces cost of annotation. But it does not matter. What matters is the saving on annotating the corpus generated from autocompletion based on this corpus.

## 3. Towards Generation

Once autocompletion starts working, we are ready for the next step: generation with the grammar used for autocompletion. As in the case of autocompletion, we stay away from the mainstream approach. The mainstream approach to generation is statistical pruning, exemplified by Langkilde (2000) and many others. Statistical pruning in generation goes hand in hand with the emphasis placed on robustness in parsing by the Natural Language Processing community today, which results in grammars that overgenerate. The idea is to generate a lot of garbage first and then clean it up. The same kind of garbage is also an issue we are facing because we do not take care of overgeneration by a handcrafted grammar but just extract all combinations of categories from the corpus used for autocompletion. We address this issue by leaving the pruning task to users. Pruning occurs every time a user selects one of the predicted phrases in a list presented to him in our authoring tool. Predicted phrases not selected by the user are pruned, leaving user input, essentially a sequence of predicted phrases provided by autocompletion as the only generated text. The generated text is necessarily parseable by the grammar used for autocompletion and the needed syntactic structure for automatic annotation of the generated text is ready after the user selects the last predicted phrase that completes a sentence. Figure 2 illustrates this generation process. The text before slashes in the figure is user input and the text after slashes is provided by autocompletion. Notice that user input is repeated in the prediction provided by autocompletion. This is necessary because a user is not allowed to freely type anything. Predicted phrases starting with the same user input are listed horizontally and linked to each other by bidirectional arrows. Pairs of user input and predicted phrases not selected by the user are crossed out. This means they are pruned.

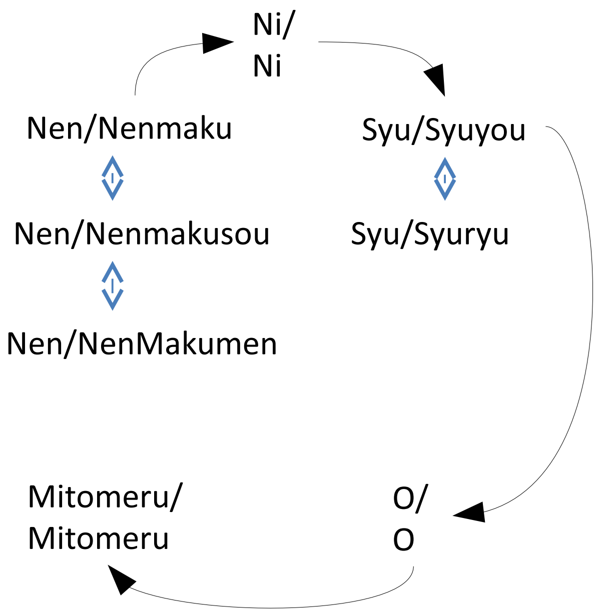


Figure 2: Generation by autocompletion

## 4. Evaluation

In order to tell the effectiveness of the process of generating annotated corpora with autocompletion outlined here, we present three pieces of data. First comes the annotation time for a report of 109 words: 32m3s. All sentences in this report are parseable by the grammar on which autocompletion is based. Next comes the measures of efficiency of autocompletion that allows input of phrases not provided by autocompletion(the way autocompletion normally works in a browser) in table 1. Finally, we give the input rate of autocompletion that disallows input of phrases not provided by autocompletion: 0.75s/char.
These data are obtained from the following experimental procedures:

1. Several physicians are asked to use our authoring tool of pathology reports with built-in autocompletion that allows input of phrases not provided by autocompletion (Only two of them who use the autocomplete function for more than 5%of the input text are considered proper subjects for evaluating the efficiency of autocompletion).

| | |
|---|---|
| length of predicted portion | 47/106 |
| input rate of predicted portion (s/char) | 0.68/1.46 |
| length of fragment not completed by autocompletion | 500/615 |
| typing rate of fragment not completed by autocompletion (s/char) | 3.98/4.29 |
| percentage of text completed by autocompletion (%) | 9.4/14.7 |
| reduced typing time (%) | 83.91/65.97 |

Table 1: Measures of efficiency of autocompletion that allows input of phrases not provided by autocompletion

2. A report by one of the subjects is rewritten by one of us without using any word not predicted by autocompletion.

3. The same report is annotated manually.

## 5. Discussion and Conclusion

Our experiment data shows that automatically generating corpora with autocompletion that limits user to selecting predicted phrases is feasible and effective in a controlled language environment. Autocompletion plays two important roles: First, it enables users to reduce a lot of time spent on inputting long words typical to our environment that can more than cancel out the extra time wasted on selecting short words from the list of predicted phrases. Even slowed down by moving their hands from the keyboard to the mouse for selecting short words, they can still achieve an overall input rate per character faster than their own typing speed. At no cost in terms of man-hour to the user, annotated corpora can be created with autocompletion. Another important role played by autocompletion is that it guides users to input parseable text such that user input that cannot be included as part of an annotated corpus is minimized. Corpora generated from our method are also guaranteed to be continuous text, making them useful for text understanding and anaphora resolution. It is impossible to generate a corpora that can serve these purposes by alternative methods like creating a corpus by running a parser on some text and including in it only unrelated parseable sentences, assuming the parser comes with an imperfect grammar written with limited investment in man-hours.

## 6. Acknowledgements

The work described in this paper is funded by Fuji Xerox Company Limited and Japan Science and Technology Agency.

## 7. References

Chamberlain, J., Poesio, M., and Kruschwitz, U. (2008). Phrase detectives - a web-based collaborative annotation game. In *Proceedings of I-Semantics*, Graz, Austria.

Karl, M. (1999). Tackling the complexity of context-free representations in example-based machine translation. In *Proceedings of the 7th Machine Translation Summit*, pages 254–263.

Langkilde, I. (2000). Forest-based statistical sentence generation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics Conference*, pages 170–177.

Nandi, A. and Jagadish, H. (2007). Effective phrase prediction. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, pages 213–230.

Nederhoff, M.-J. (1993). Generalized left-corner parsing. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 305–314.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast —but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263.

Wood, M. (1996). *Syntactic Pre-Processing in Single-Word Prediction for Disabled People*. Ph.D. thesis, University of Bristol.

# Developing Lexical Resources for Controlled Authoring Purposes

**Kara Warburton**
City University of Hong Kong
Tat Chee Ave., Kowloon, Hong Kong
E-mail: kara@termologic.com

**Abstract**

Controlled authoring is increasingly being adopted by global organisations as a means to improve their content and manage translation costs. In recent years, some controlled authoring software applications have become available. These applications assist writers in adhering to rules of style, grammar, and terminology. In this paper, the author describes her experiences in implementing lexical resources in a controlled authoring software. Incorporating pre-existing terminological resources presents challenges because they were typically developed as an aid for translators. While there are economies of scale to be achieved by addressing the needs of controlled authoring and translation from one lexical resource, careful planning is required in order to account for different requirements.

**Keywords:** controlled authoring, terminology, word usage.

## 1. Introduction

Controlled authoring is "the process of applying a set of predefined style, grammar, punctuation rules and approved terminology to content (documentation or software) during its development" (Ó Broin, 2009). In this paper, we describe some approaches for addressing the terminology requirements for controlled authoring.

## 2. Uses of controlled authoring

Controlled authoring (CA) is increasingly being recognised as a means to improve content quality and overall communications in a company or organisation. Content that is consistent and easy to understand is more effective at achieving its objectives, i.e. increase sales in the case of marketing material, improve product usability in the case of product information, and deliver more effective public services in the case of public institutions. It is also easier and less costly to translate. Organisations that produce significant quantities of information content, especially if they do so in multiple languages, can therefore realise significant benefits by implementing controlled authoring.

## 3. Active and passive controlled authoring

Controlled authoring can be either active or passive. Passive controlled authoring refers to the cases where the organisation provides a style guide and a word usage list to its writers, who refer to these materials when they feel so inclined as they work. Active controlled authoring leverages a computer application to prompt writers to adopt style rules and recommended words as they write. Active controlled authoring guarantees, more or less, that the rules are followed whereas passive controlled authoring relies on the voluntary initiative of the writers. Often, writers are not even aware when a style rule or vocabulary choice applies to a text that they are writing and therefore they may not even look at the authoring guidelines when they need to. In this paper, we focus on active controlled authoring.

## 4. Controlled authoring software

Examples of software applications that are used for active controlled authoring include Acrolinx [1], crossAuthor [2] (Across) and HyperSTE[3] (Tedopres). Much of the material in this paper is based on the author's experiences implementing and using lexical resources in Acrolinx. The Acrolinx software is configured for an organisation by a designated Administrator, who defines the style and grammar rules to be used, incorporates organisation-specific vocabularies, and sets up other functions in the system such as reporting.

## 5. Lexical resources for CA

Most definitions of controlled authoring, like the one cited earlier, refer to "terminology." However, this is not entirely accurate. The vocabularies used in a CA application differ from conventional terminology resources in two ways. First, many of the lexical items required for CA are words and expressions from the general lexicon. For example, a company may prefer writers to use the adverb *almost* instead of *nearly*, the latter potentially being confused with a spatial concept. According to convention, such items would not be included in a terminology resource, since "terms" are domain-specific. Second, CA applications have a greater need for verbs, adjectives, adverbs and even some prepositions and other function words than are normally found in terminology resources. This was confirmed in a study of one company recently carried out by the author (Warburton 2014), where only 31 percent of the lexical items required for controlled authoring were nouns. In contrast, it is generally agreed that about 90 percent of the terms found in any given termbase are nouns. For these reasons, we prefer the term "lexical resources" to refer to the linguistic units required in the dictionary or "terminology" function of a controlled authoring tool.

On the other hand, by using the term "lexical resources" for

[1] www.acrolinx.com

[2] www.across.net

[3] www.tedopres.com

CA, we do not suggest that they are lexicographical, structurally speaking. On the contrary, CA resources need necessarily adopt the concept-based data model and entry structure used in terminology and which are defined in ISO 16642 (Terminology Markup Framework -TMF) and ISO 30042 (TermBase eXchange - TBX). This is because organising the individual words and expressions into synonym sets, or synsets, is fundamental to CA. For each prohibited word there is a preferred alternate, for each acronym there is an expanded form to be used on first occurrence.

## 6. Synsets

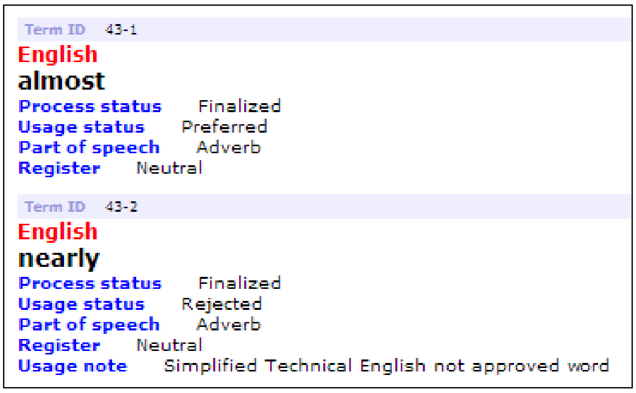
In CA, the goal is to address the problem where different people are using different words for the same thing. The aim is to adopt a “controlled vocabulary.” The first step is identifying cases where this is occurring and the expressions involved. Each case represents a synset, which is recorded in a lexical database using the TMF model. Each expression is then given a usage status value.

The following is an example of a synset entry for the words mentioned earlier, in TBX format, that would be suitable for a CA application.

```
<termEntry>
 <langSet xml:lang='en'>
  <tig>
    <term>almost</term>
    <termNote type="partOfSpeech">
    Adverb</termNote>
    <termNote type="Usage_Status">
    Preferred</termNote>
    <termNote type="Register">
    Neutral</termNote>
  </tig>
  <tig>
    <term>nearly</term>
    <termNote type="usageNote">Simplified
Technical English not approved word</termNote>
    <termNote type="partOfSpeech">
    Adverb</termNote>
    <termNote type="Usage_Status">
    Prohibited</termNote>
    <termNote type="Register"
    Neutral</termNote>
  </tig>
 </langSet>
</termEntry>
```

Note the properties of this entry. Each term or lexical unit is expressed in a <tig> (term information group) along with two key other descriptors: the part of speech, which serves to disambiguate homographs, and the usage status, which differentiates preferred and prohibited terms. In the section of the prohibited term, a usage note is also provided as a justification. In the above sample, the company terminologist has chosen to use a neutral register descriptor to identify entries in the central termbase that are intended for the controlled authoring software. As we shall see later, this is a wise approach since not all terms in a termbase are useful in a CA application.

In addition to delivering this information to content producers through the CA application, the organisation can also make it available on a Web site for virtually any employee. This approach provides for both active and passive controlled authoring. Figure one shows a screen capture of the previous entry from TermWeb (Interverbum) [4], a web-based terminology management system.


Term ID 43-1
English
almost
Process status Finalized
Usage status Preferred
Part of speech Adverb
Register Neutral

Term ID 43-2
English
nearly
Process status Finalized
Usage status Rejected
Part of speech Adverb
Register Neutral
Usage note Simplified Technical English not approved word

Figure 1: Synset shown in TermWeb

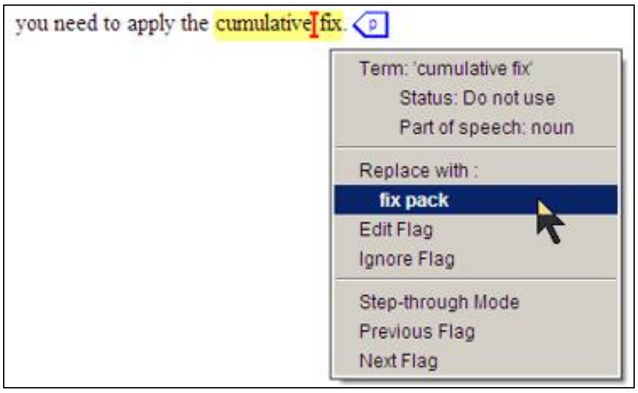
## 7. Usage status values

In the first implementations of active CA, it was thought that the two usage status values that we have just seen (prohibited and preferred) were sufficient, and some CA applications are restricted to these two values still today. Indeed, the CA specification for the aerospace industry, Simplified Technical English[5], which has seen uptake in other industries, includes only these two values. However, some organisations require additional values to handle cases that are not so clear-cut, particularly, restricted terms and admitted terms. A restricted term is prohibited in some contexts but allowed in others. An admitted term is allowed, while not being the preferred choice.

The CA application uses these values to prompt writers with guidelines on vocabulary use, a function sometimes referred to as term checking. If a writer uses a prohibited word, it will be visibly highlighted. The writer can then access information about the preferred alternate. In order for this function to work, the words and terms in the back-end lexical database have to be marked with appropriate usage values. Figure two shows the result when the writer right-clicks on a prohibited term, here highlighted in yellow.

[4] www.interverbumtech.com/

[5] www.asd-ste100.org



you need to apply the cumulative fix.

Figure 2: Term checking

Restricted terms require special attention. First, as previously mentioned, not all CA applications can handle this usage status. Restricted items should be visibly highlighted in a manner that easily distinguishes them from prohibited terms. The user should be able to access information about the conditions of restriction, such as in which subject field or product areas the term is disallowed. This requires more information in the back-end database, such as subject field values, product values, and usage notes. It is not possible to show usage advice in the form of long prose in the limited space of context menus such as the one above. Such descriptions can be included in additional help files, which can be accessed from the context menu by double-clicking on the preferred term (here, *fix pack*). This action opens a new window where more information can be provided. However, to maximise usability, one should aim to include as much information as possible *should* on the context menu. This leads to certain design choices, such as using a limited set of usage note values to replace some of the prose descriptions, and also, keeping the usage notes themselves as succinct and to the point as possible.

Acrolinx allows error types to be given "penalty" scores. This is useful for compiling metrics about writing quality and to track improvements over time. Here, restricted terms should not have the same penalty score as prohibited terms as clearly, the error is less severe. Similarly, writers should not be penalised for using admitted terms. In some cases, an organisation may decide to disable the penalty scores for errors in vocabulary choice altogether, recognising that writers should not be expected to know about potentially hundreds if not thousands of company preferences for specific words. One has to be careful not to discourage writers with negative performance reports.

## 8. Incorporating pre-existing resources

Lexical resources are essential for effective CA and therefore organisations that have a pre-existing terminology database (termbase) when they initiate CA are considered to be at an advantage compared to those that have to build lexical resources from scratch. However, there are challenges incorporating an existing termbase into a CA application.

First, having probably not been developed with CA in mind, the termbase may lack some of the necessary descriptors. Most termbases, originally developed as an aid for translators, contain few synsets, and any existing synsets are unlikely to have usage status values; knowledge about synonyms and their relative importance in the source language is not of great interest to translators. It is therefore necessary for the company or organisation to retrofit the termbase with synonyms in the source language, usage status values, usage notes, and other information of use to content authors. This can take a considerable amount of time.

Another consideration is whether or not all the existing terminology should be in the CA application. While having company-specific terminology in the CA tool is normally beneficial for word recognition and spell checking, there can also be some conflicts between the so-called "terms" and the other expressions needed for CA. For instance, the term *readme* may exist in the termbase in several different entries: (1) playing the grammatical role of a modifying adjective, as when used in "readme file," and (2) noun, as when used in "the readme." The noun entry, intended to be used in the CA tool, comprises a synset with two terms: *readme* as a prohibited term and *readme file* as a preferred alternate. Provided that the CA tool strictly takes into consideration the part of speech values from the termbase, there should not be any conflict between these two entries. However, CA tools sometimes assign their own part-of-speech values to entries, using morphological analysis which works in conjunction with the syntactical analysis that operates on running text in order to produce matches between terms in the text and terms in the database. In this case, the adjective part of speech value of the first *readme* entry may be overwritten by the noun value assigned through morphological analysis, leaving now two entries for *readme* having the same part of speech value. As a result, the usage advice to avoid using readme as a noun may not be retrieved during term checking. If the company's existing termbase is large, there may be many such conflicts, particularly for a language such as English where homographs are common.

For spell checking purposes, the CA tool has its own internal spelling dictionary for general language. However, words that are unique to the company, such as product names, will not be in the internal dictionary and will therefore be flagged as a spelling error, resulting in a false positive. To reduce this problem, a secondary spelling dictionary can be added to the CA tool. However, if the secondary dictionary contains words that are already in the internal dictionary, performance can be impacted. It is therefore necessary to remove from the secondary dictionary any words that are already in the internal dictionary. Of course, acronyms and abbreviations should also be excluded from the spelling dictionary as they also produce many false positives.

As a consequence, the company or organisation needs to evaluate its termbase entries to determine which ones are effective in the CA tool, assign the necessary descriptors, and test the entries to verify that they work properly in terminology checking. Testing involves seeding a text with the terms in various syntactic positions (noun, verb, adjective), alone and in various compounds. This is the only way to get predictable results; simply importing an existing termbase into a CA tool causes more damage than good. In our experience, depending on the size of the original termbase of course, only a fraction of the total existing entries are beneficial.

## 9. Conclusions

In this paper, we demonstrated that the lexical resources required by CA applications are not "terminologies" in the conventional sense; a much broader range of lexical items are required. Synsets with usage status values are absolutely essential for term checking, and there is a need to manage more than just "prohibited" expressions. Lexical units presenting unique surface form characteristics are needed for spell checking. Pre-existing terminology resources cannot be directly incorporated into a CA application without significant customisation and modification.

## 10. References

Ó Broin, Ultan (2009). Controlled Authoring to Improve Localization. *Multilingual*, Oct/Nov 2009. Multilingual Computing Inc.

International Organization for Standardization (2008). *ISO 30042: TermBase eXchange (TBX)*.

International Organization for Standardization (2003). *ISO 16642: Terminological Markup Framework (TMF).*

Warburton, Kara (2014). *Narrowing the Gap Between Termbases and Corpora in Commercial Environments*. PhD Thesis, City University of Hong Kong.

# Marrying Technical Writing with LRT

**Giovanni Antico, Valeria Quochi*, Monica Monachini*, Maurizio Martinelli****

S.Te.L S.r.L, Istituto di Linguistica Computazionale - CNR*, Istituto di Informatica e Telementica - CNR**

giovanni.antico@stel-web.it, name.surname@ilc.cnr.it*, maurizio.martinelli@iit.cnr.it**

## Abstract

In the last years the Technical Writer operational scenarios and the workflow sensibly changed; specifically,"free style" writing - or manual writing - has become outdated and technical writing is now much more concerned with structured management of content than in the past. Technical writing has become more demanding due to a number of factors among which the rise and spread of mobile devices usage. This paper discusses the new needs of technical writing and content management business and how LRT can help it improve quality and productivity.

**Keywords:** controlled language, technical writing, content management systems

## 1. Introduction

In the last years the Technical Writer operational scenarios and workflows sensibly changed; specifically,"free style" writing - or manual writing - is becoming outdated and technical writing much more concerned with structured management of content than in the past.

Actually, technical writing has become more demanding due to a number of factors including the increased complexity of machines, which translates into more complex documentation, and the rise and spread of mobile device usage, which generate new information needs in users. Additionally, the legislation is now much more compelling and standardization of technical documentation more widespread. All these aspects makes writing technical documentation increasingly more complex as there are many different requirements to be fulfilled.

Karen McGrane, a pioneer in Content Strategy, states the need to start thinking at content as independent of its presentation means and to structure content so that it can be reused"(McGrane, 2012).

This view of structured content not as a mere technological matter, but as strategic for business today is becoming an acknowledged reality. Content needs to be decoupled from (form of) presentation and made modular, reusable and collaboratively modifiable according to rigorous workflows that will help business keep up with the increasing need of adapting the same content to different presentation devices.

In addition to these display issues, technical content must also satisfy a number of quality requirement among which, crucially, coherence and comprehensibility. Standardization, controlled languages, and language simplification are key issues here.

Content Management Systems (CMS) have started to be used in the technical writing industry and are successfully fulfilling some of the new requirements, especially related to modularization of content, collaboration and sharing among colleagues and in some cases beyond.

Most needs related to content creation and quality however, still fall outside the scope of existing CMS, which mainly address optimal management of document structure and work-flow management[1]. At least part of such needs and of the desiderata for for technical documentation within CMS, we claim, can be satisfied with the help of current state-of-the art Language and Resource Technology, especially exploiting the web-service paradigm (cfr. platforms such as PANACEA[2], OPENER[3], Let'sMT[4]).

This contribution will attempt to identify some of those pressing needs and the language-related technologies that might provide an answer to them.

### 1.1. Information needs

We can identify two types of information needs: general needs, i.e. always valid and already well-known, and emergent ones.

It is well-known and fundamental for content to be first of all correct, updated and coherent within the whole set of documentation that goes with the industrial product along its whole lifecycle (i.e. from its commercial proposal, to its implementation, client education, billing, maintenance and repair, etc.). Technical documentation has to be easy to understand by clients' workers, and therefore written in the user language (cfr. (O'Keefe and Pringle, 2012) and (Laan, 2012)).

On top of these, the recent "web" and "T" revolution has created new needs that technical writing businesses have to address (McGrane, 2012). Content today needs to be:

- multimodal, i.e. documentation has to be based on images and video in order to facilitate the comprehension of the sequences of tasks to perform;
- searchable;
- contextual, that is content has to be retrieved at the right place in the right moment, whereas at present content is usually placed elsewhere than needed;
- targeted at users' profiles in order to avoid information overload;

[1] e.g. Argo CMS, www.keanet.it/; Vasont CMS, www.vasont.com; docuglobe,www.gds.eu/; SCHEMA ST4, www.schema.de/,...

[2] www.panacea-lr.eu

[3] www.opener-project.org

[4] www.letsmt.eu

- equipped with tools for sharing problems and solutions;
- integrated and aggregated (dynamically); i.e. users should not need to consult different information sources to retrieve the data they need.

### 1.2. Standards for Technical Writing and controlled languages

Legislation is vast in this field both at the national and international level and aims at regulating both the planning/design of the products and the instructions on their usage for security and quality reasons (e.g. ISO-IEC-82079-1 (2012); UNI-10653 (2003); ; to mention just a few).

In addition to these normative rules, we find a number of standards and best practices adopted more or less widely, such as the AECMA / ATA / S1000D for the aerospace technical domain (ACEMA/ASD-S1000D, ); the OASIS DITA for e-business applications (OASIS-DITA, 2010); and various best practices for technical writing.

All these aim at improving the quality not only of content itself, but also of processes for editing, translating, publishing and disseminating technical documentation, by ruling the work flow as well as the structure, presentation order and informativeness of documents, the semantic communication rules, their graphical display, the file format and many other aspects.

As it can be easily imagined, free-style writing is error-prone in coping with all the requirements imposed by legislation and best practices. Think for example at a simple case: a warning. Legislation requires that a warning is accompanied by a pictograph, has a label that explains the type of warning (attention, danger, prohibition, . . . ); the cause, consequences and remedies are explained. Dealing with content manually, the technical writer needs to recall and apply the correct structure, paginate the image, and assign the correct style to each piece of content.

Using a CMS (Content Management System) as a control software instead allows for the definition and automatic application of the required structure and for the automatic insertion and editing of the image. The CMS can also export the same content in various file formats, especially in the XML targeted standards, like for example DITA.

## 2. Advantages of CMS for technical writing

Given the strategic importance of smart structured content management, CMS have become widely used in the industry, with current systems successfully satisfying many of the needs mentioned above and brings a number of advantages to content business. In the following, we mention the most salient.

**Collaborative management of content** With an adequate and customisable management of profiles and authorization, CMS allows different professional figues, both within and outside the company (e.g. product manager, technical writers, translators, consultants, etc.), to collaborate to the process of content editing according to predefined work flows.

**Single management and revision of content that can be reused** CMS allows unique management of content creation and revisions and makes the various pieces of content reusable at all levels. For example, sections, chapters, warnings, variables, . . . , are handled as independent units of content that can be reused or visualised in different contexts and/or displayed differently depending on the dissemination/presentation channel chosen. Moreover, content managers can choose whether a collaborative revision of common content should be propagated automatically to all its instances (i.e. all occurrences in the various documents) or not.

**Definition and application of content structures** Reusable models or templates for different kinds of documents, for examples for the warnings mentioned above, can be defined and used to help writers in their daily work. This reduces errors and costs by increasing efficiency.

**Automation of the production of various types of targeted technical documentation for different channels** By integrating automatic pagination tools with web applications, CMS is be able to automate cross-media publishing functions.

This directly translates into several advantages for the business:

- production costs and time will be reduced,
- content correctness and quality will increase,
- information will be easily targetable,
- graphical styles can be made more coherent.

**Content Tagging for reuse** Faceted tagging or classification, possible in CMS, allows for a quick ordering and filtering of content according to several different characteristics or points of view and constitutes a sensible improvements relative to taxonomic classification. By means of tagging, the technical writer defines the usage context of a piece of common content; that is, for example, (s)he defines:

- which family, model, machine or Bill of Material it refers to,
- who is its audience,
- what types of publication it was conceived for,
- what types of presentation/display channel it can be disseminated on,
- . . .

Tagging and classification within existing CMS however is still manual.

**Translation Management** Through a tagging system, a CMS makes it possible to provide a translator with only the bits of content to be translated of proofread, this helps in optimising the translation costs. Moreover, translations via CAT tools can be automatically imported in the CMS, thus saving time and gaining accuracy in text alignment, which can be totally independent of the technical writer language skills. However, heavy manual intervention/work is still needed for managing translations and technical writers still need to use different software applications.

**Integration** CMS often shares data (classifications, codes, prices, figures . . . ) with other software applications used by the company for: e.g. project management, CAD, ERP, . . .

## 3. What's missing

Figure ( 1 ) exemplifies a typical working methodology of a technical writing company.

After a preliminary analysis of the documentation project to be realised, the rules that characterise the documents are defined and fixed by the writing and editing team. At this point (Author phase), the authors start creating new content or insert missing bits of data into the database.

The next phase, the selection, allows defining the specific document that will be created through a process of automatic pagination.

If we take this example of (real-world) working method, we can see that CMS solutions normally handle two of the indicated phases: the author and selector phases in Figure (1).

However, while the selector phase can be highly controlled/structured, the author phase is generally "free". In the selector phase, in fact, the CMS allows for the definition of several (formal) rules (customisable on project basis) that make sure documents respect a number of fundamental requisites: for example, that there cannot be a picture without a caption, or that there cannot be a sub-paragraph if there is no preceding paragraph, etc. In the author phase, i.e. the writing phase, instead, current solutions offer little or no support to authors[5].

### 3.1. Desiderata for Technical Documentation CMS

**Content Tagging - metadata management** Dynamic adaptability in context is required for dealing with different domain terminologies: e.g. a technical writer would need to use different terms when writing about a system for maritime navigation than when writing about a similar system on an automobile (*route* vs. *direction*). Current systems rely on manual tagging and on manual metadata choice. Some kind of automation of support here is desired for improving content adaptability and for reducing errors.

**Glossaries and terminologies** Writing technical documentation requires the availability of technical glossaries and terminology to help writers and editors be consistent and clear. Thus, existing glossaries and terminologies should be integrated in the authoring/editing phases within CMS in order to offer better support to content managers and technical writers. In addition, as often glossaries and terminologies need to be customised according to the specific project, tools that help such a collaborative customisation of terminologies would be most welcome.

**Advanced Translation Management** Translation is currently often outsourced, done by professional translators externally of the CMS, i.e. of the "routine" working environment. This clearly increases not only the time and costs of the final documentation, but also the rate of human errors. Ideally, technological solutions that help or assist professionals in translating technical documentation should be integrated into the CMS that handles all other phases of the workflow, so that the whole is more efficient and controllable, by the editor or the project manager.

**Advanced Integration** A more thorough integration of the CMS with other software can be highly advantageous for companies. It would indeed allow its different sectors and collaborators to: share correct information, to use it to write the documents that accompany the products along their life cycle, and above all to dispose of complete information at once, without having to consult different sources to get the complete picture required.

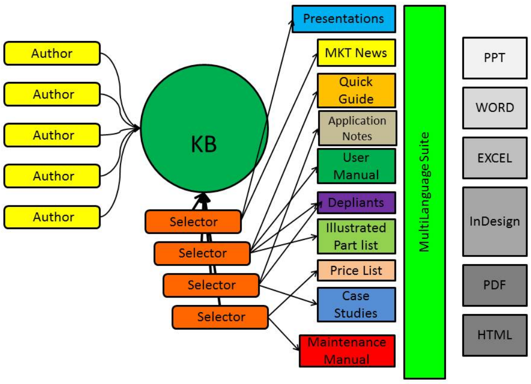


Figure 2: Work flow

**Controlled and Simplified Language** Current CMS and technical writing tools, as we have seen above, mostly deal with formal issues related to the segmentation of content into minimal reusable pieces, collaboration and sharing among the working team, and display and presentation of the content on different media. Aspects related to the quality of the content, i.e. on the information conveyed, however, are still for the large part left to the human writer to control. However, looking at the current landscape we see that the adoption and use of controlled (natural) languages has become now a best practice in many sectors. Simplified Technical English (STE[6]) is for instance commonly used now in the editing of technical documentation, especially within the aerospace and military industry[7].

Controlled (natural) languages generally restrict the grammar and vocabulary of the language in order to reduce or eliminate ambiguity and complexity (normally a word can have only one sense and one part-of-speech). For example: *close* can be used as the verb denoting the action of closing,

[5] Current CMS solutions can provide standard spell-checking functionalities, but little or no advanced linguistic or terminological support.

[6] www.asd-ste100.org/

[7] Other known controlled languages used in the industry are Caterpillar Technical English, IBM's Easy English, BULL Global English, . . .

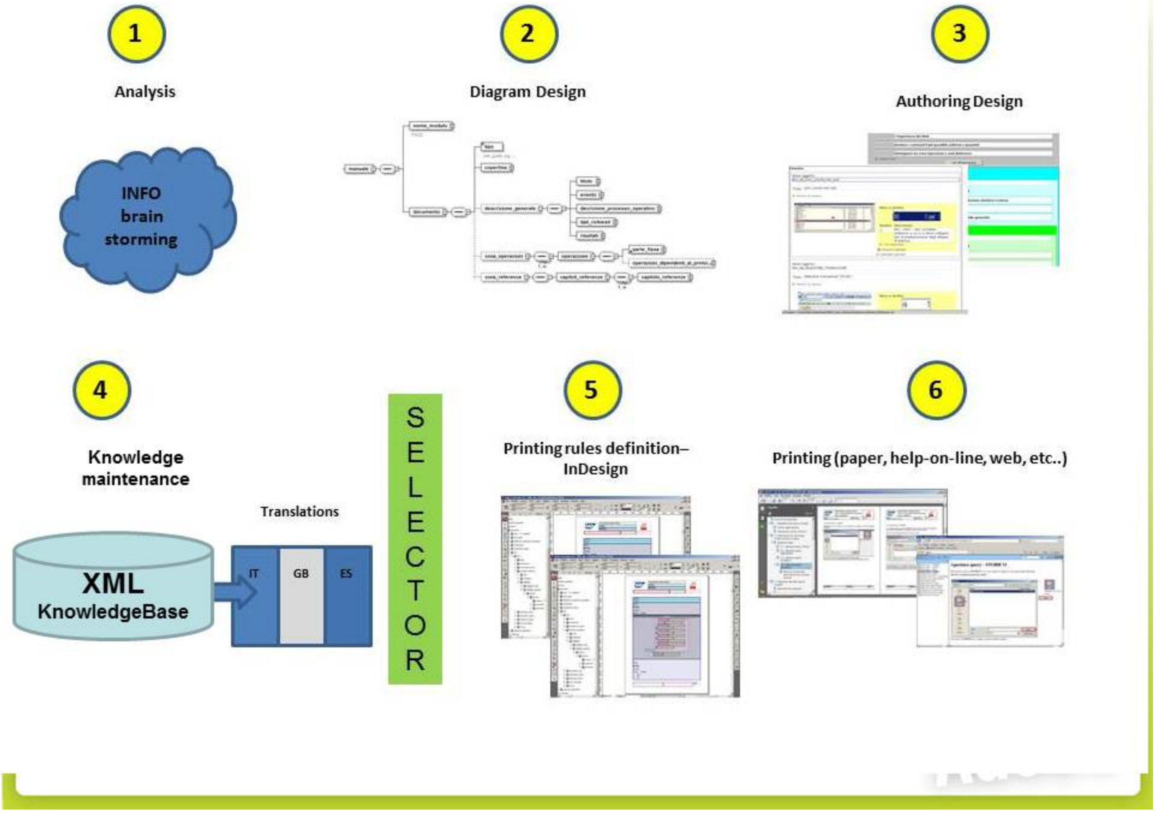


Figure 1: A Working Methodology

but not as the adjective for proximity; this way, to *close the door* is accepted as a valid chunk in the controlled language while *do not go close to the landing gear* is not.

The advantages of the adoption of a controlled language is now widely recognised in the sector: it helps increasing clarity of the procedural technical language; it also helps improving the comprehension of the documents by non-native as well as low-literacy speakers of the language and optimizing translation procedures by increasing the performance/reliability of CAT and MT tools.

To adopt a controlled language, however, forces the technical writer to follow the prescribed (language) rules and restricted vocabulary specific for the topic/domain of the documentation, which increases the complexity of his/her, and the editor's, job.

Therefore, equipping a CMS for technical documentation with smart authoring tools that support the adoption of controlled languages would bring together their potentialities to the advantages offered by structured management.

It is worth noting, however, that for many languages and domains, standardised or shared controlled languages do not (yet) exist; each company or manufacturer simply define their own requirements and establish some internal best practice (often even with rules not explicitly formalised or stated). Moreover, even when a CL exists, technical writing projects often need specific stylistic rules and terminology that goes beyond the standardised controlled language. In such cases, CMS could/should additionally provide functionalities that help writers and editors establish their in-house solutions of controlled language (rules and vocabulary), which may be additionally shared to a wider community in order to contribute to the spread and harmonisation of technical style within a single national/regional language.

## 4. How can LRT and Semantic Web help

Although human intervention in technical writing will continue to have play a major role, language technology can help optimize the writing and editing tasks considerably, as the few existing products demonstrate. Of course, more research is required to fruitfully apply the developments achieved within the research community, make the products efficient and widely available, at low cost. However, it is high time we transferred some of the more stable and mature technology to the small industry for exploitation. LRT advancements can help in particular address desiderata related to glossaries and terminologies, Metadata management and domain adaptability, translation management, and support for controlled or simplified language.

Technology today is relatively mature to bootstrap lexicons, terminologies and ontologies from corpora, and merge or link them to construct resources that cover different domains and usages (cfr. Venturi et al. (2009), Carroll et al. (2012), Fazly et al. (2007), Lin et al. (2009), Del Gratta et al. (2012), Padró et al. (2013) among many others). Of course, these are still error-prone procedures; but together with automatic reliability scores for semi-automatic validation and post-editing tools, they may help reduce the production costs and increase the coverage of corporate resources.

Thanks to the effort of the computational linguistics community in the last decades, we now have available (standardized) representation formats and architectures for lexical, terminological and ontological resources, which allow for easy integration within various applications[8]. Also a number of such resources already exist available on the web[9], which can be used already to build prototypes and showcases (e.g. WordNets, Ontologies, Lexica (Francopoulo et al., 2009), (Henrich and Hinrichs, 2010), (Del Gratta et al., Under review) ). There has been also a great body of work on metadata management and standardization for terminology management, which is now largely adopted especially in the translation and localization world, but not so much in technical writing, and not so much for languages other than English.

Exploiting such representation models and standardized metadata, resources can be organized to list different usages for different domains/context, so that once the domain is identified or selected they can be used for providing some automation in content tagging and adaptability.

Terminologies and ontologies can also be further adapted to manage controlled vocabularies by applying for example automatic acquisition and representation of term variants and acronyms (e.g Jacquemin and Tzoukermann (1999), Thompson et al. (2011)), word sense induction and classification (e.g. Lau et al. (2012), Manandhar et al. (2010), Pantel and Lin (2002)) , such that the preferred term can be automatically suggested when a synonym or variant is used.

Support to authoring (in a controlled language) and copy editing can be provided by parsing tools, which can be adapted and used to signal complex word patterns and syntactic structures. At a simplest level, the integration of a pos-tagger (which nowadays exist state-of-the-art for many languages) in a technical authoring system can be used, for example, to signal and highlight not allowed part-of-speeches, e.g. adverbs; the integration of morphological or syntactic analysers can help identify for example the usages of the passive voice, which should be revised/changed by the author. Nonetheless, more sophisticated tools for checking whether the sentence the author is producing is compatible with the syntax of the controlled language (assuming this has been defined) can also be developed building on the existing technology.

Furthermore, domain-specific authoring memories could be implemented, and potentially shared across companies, that collect previously used (and/or approved), chunks of texts, so that within the CMS, the author is provided (in real time) with suggestions about the most similar and or frequent chinks of text already used within the domain, the same document, or the same company documentation material. Finally, as technical documentation often needs to be written in the language of the target audience, and translation from English is not always an option (in the first place because documentation may be originally written in another language), Machine Translation systems as well as translation memories should be integrated in CMS and adapted to suit the needs of technical writers and editors. This way, massive outsourcing of the translation process can be avoided while at the same time editors or project managers can be given higher control on the whole workflow. CMS enriched with automatic MT should further be equipped with post-editing applications, so that the revision of content becomes cheaper. Instead of having independent translation tools, be they fully automatic or not, what is needed today especially by SMEs in small countries (i.e. with smaller markets), is to provide the industry with easy-to-use machine and /or assisted translation tools for many language pairs. Also of interest is the design and deployment of shared translation memories that are capable of incremental and smart augmentation, so that human translation can become easier and ensure coherence within a same domain, topic and language style.

Technical authoring tools that implement some of the functionalities mentioned above do indeed exist already (e.g. Acrolinks IQ Suite[10], Boeing Simplified English Checker[11], Adobe Technical Communication Suite); most of them however are developed and marketed as legacy toolkits by big industries, at prices that SMEs in countries where the internal market is not very big cannot afford. Even more importantly, perhaps, they support few EU languages, mostly English and German. Very little exist for other languages[12].

To promote competitiveness in non-English speaking countries, instead, such technology needs to be mastered also by smaller companies that can adapt and customise solutions for their specific reality. In recent years, language technology has been looking to the web and the paradigm of language tools as (distributed) web-services and web applications is now relatively consolidated. This allows for modularisation, easy experimentation by companies, better academia-industry transfer, and is in line with the "Software as Service" paradigm currently adopted by many CMS software houses. Thus, while the academia should disclose its achievements as open source software, it is also important to pursue with research on deployment of language technology in the web-service paradigm, so that new functionalities can become quickly and easily usable by businesses not interested in technology development *per se*.

## 5. Conclusions

Given the recent change in paradigm and strategy for technical writing business, and that information is a very important part of products, satisfying all information needs of

[8] e.g. LMF, NIF, TMF, Lemon, among others, which can all be represented in XML, RDF, RDF Linked Data, or Json syntax cfr. Bora et al. (2010), Hayashi et al. (2012), McCrae et al. (2012)

[9] We will deliberately not consider licensing issues in this discussion.

[10] www.acrolinx.com/

[11] www.boeing.com/boeing/phantom/sechecker/

[12] In France for example a movement has started to promote the adoption of controlled technical languages and to develop some support computational tools, e.g. see the projects Sense Unique, `//tesniere.univ-fcomte.fr/sensunique.html`, LiSe (Renahy and Thomas, 2009)

users and legislation provides the product with a considerable competitive advantage. This implies that content companies have to enhance not only their publication channels and platforms, but also their content creation methodology, introducing advanced content technology support to help them go beyond "free style" writing, as part of their business strategy.

In this paper, we went through the technological features that are already available to technical writers and editors. In particular, with the adoption of CMS great improvements have been introduced both for managing the technical documentation team and workflow, and for better modularizing and structuring content so that most formal aspects of technical documentation creation and publishing are automatized.

Still, the authoring part of the job is left uncovered within CMS and authors need to recur to independent authoring software, which is highly expensive and often available for a limited se t of languages. Thus, we have formulated some of the most urgent desiderata for CMS dedicated to technical writing and have tried to clarify what LRT can provide support and briefly sketched how.

Language and Web technology can indeed help develop such a forward-looking advanced strategy. Indeed, there exists already authoring toolkits that tackle (some) of the mentioned needs. However, these are generally quite expensive for small companies, and mostly work for English and for a few highly restricted technical domains. Instead, what is envisaged is easy-to-use, customize and integrate software/services for potentially all EU languages and various text styles, so that even smaller companies can afford to introduce important innovations within their work environments.

Certainly some of the text analysis, terminology extraction and management technology is mature enough for integration into legacy applications on a web-service basis. In particular, what we try to encourage is research towards a strong integration of LRT and Semantic Web functionalities within CMS for the technical documentation business, in such a way that they can be able also to handle controlled languages in the authoring phase. This might, in fact, provide the business with a double advantage: both on the form and content sides.

## 6. References

ACEMA/ASD-S1000D. (). *International specification for technical publications using a common source database*. ASD.

Bora, S. Ali, Hayashi, Yoshihiko, Monachini, Monica, Soria, Claudia, and Calzolari, Nicoletta. (2010). An lmf-based web service for accessing wordnet-type semantic lexicons. In *Proceedings of the 2010 Language resources and Evaluation Conference*.

Carroll, John, Koeling, Rob, and Puri, Shivani. (2012). Lexical acquisition for clinical text mining using distributional similarity. In Gelbukh, Alexander, editor, *Computational Linguistics and Intelligent Text Processing*, volume 7182 of *Lecture Notes in Computer Science*, pages 232–246. Springer Berlin Heidelberg.

Del Gratta, Riccardo, Frontini, Francesca, Monachini, Monica, Quochi, Valeria, Rubino, Francesco, Abrate, Matteo, and Duca, Angelica Lo. (2012). L-leme: An automatic lexical merger based on the lmf standard. In *Proceedings of the LREC 2012 Workshop on Language Resource Merging*, pages 31–40.

Del Gratta, Riccardo, Frontini, Francesca, Khan, Fahad, and Monachini, Monica. (Under review). Converting the parole simple clips lexicon into rdf using the lemon model. *Semantic Web Interoperability, Usability, Applicability (SWJ)*, http://www.semantic-web-journal.net/system/files/swj487.pdf(1).

Fazly, Afsaneh, Stevenson, Suzanne, and North, Ryan. (2007). Automatically learning semantic knowledge about multiword predicates. *Language Resources and Evaluation*, 41(1):61–89.

Francopoulo, Gil, Bel, Núria, George, Monte, Calzolari, Nicoletta, Monachini, Monica, Pet, Mandy, and Soria, Claudia. (2009). Multilingual resources for nlp in the lexical markup framework (lmf). *Language Resources and Evaluation*, 43(1):57–70.

Hayashi, Yoshihiko, Bora, Savas Ali, Monachini, Monica, Soria, Claudia, and Calzolari, Nicoletta. (2012). Lmf-aware web services for accessing semantic lexicons. *Language Resources and Evaluation*, 46(2):253–264.

Henrich, Verena and Hinrichs, Erhard W. (2010). Standardizing wordnets in the iso standard lmf: Wordnet lmf for germanet. In *COLING*, pages 456–464.

ISO-IEC-82079-1. (2012). *Preparation of instructions for use – Structuring, content and presentation – Part 1: General principles and detailed requirements*. ISO.

Jacquemin, Christian and Tzoukermann, Evelyne. (1999). Nlp for term variant extraction: Synergy between morphology, lexicon, and syntax. In Strzalkowski, Tomek, editor, *Natural Language Information Retrieval*, volume 7 of *Text, Speech and Language Technology*, pages 25–74. Springer Netherlands.

Laan, Krista Van. (2012). *The insiders guide to Technical Writing*. XML Press, Laguna Hills, CA.

Lau, Jey Han, Cook, Paul, McCarthy, Diana, Newman, David, and Baldwin, Timothy. (2012). Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 591–601, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lin, Jimmy, Murray, G.Craig, Dorr, BonnieJ., Haji, Jan, and Pecina, Pavel. (2009). A cost-effective lexical acquisition process for large-scale thesaurus translation. *Language Resources and Evaluation*, 43(1):27–40.

Manandhar, Suresh, Klapaftis, Ioannis P., Dligach, Dmitriy, and Pradhan, Sameer S. (2010). Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 63–68, Stroudsburg, PA, USA. Association for Computational Linguistics.

McCrae, John, Montiel-Ponsoda, Elena, and Cimiano, Philipp. (2012). Integrating wordnet and wiktionary with *lemon*. In *Linked Data in Linguistics*, pages 25–34.

McGrane, Karen. (2012). *Content Strategy for Mobile*. http://www.abookapart.com/products/content-strategy-for-mobile.

OASIS-DITA. (2010). *Darwin Information Typing Architecture (DITA)*. OASIS.

O'Keefe, Sarah S. and Pringle, Alan S. (2012). *Content Strategy 101: Transform Technical Content Into a Business Asset*. Scriptorium Publishing Services, Inc., Durham, NC.

Padró, Muntsa, Bel, Núria, and Necsulescu, Silvia. (2013). Towards the fully automatic merging of lexical resources: A step forward. *CoRR*, abs/1303.1929.

Pantel, Patrick and Lin, Dekang. (2002). Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 613–619, New York, NY, USA. ACM.

Renahy, Julie and Thomas, Izabella. (2009). Compagnon lise: A collaborative controlled language writing assistant. In *ISMTCL Proceedings*, Bulag, pages 223–230, Besancon, France. PUFC.

Thompson, Paul, McNaught, John, Montemagni, Simonetta, Calzolari, Nicoletta, del Gratta, Riccardo, Lee, Vivian, Marchi, Simone, Monachini, Monica, Pezik, Piotr, Quochi, Valeria, Rupp, CJ, Sasaki, Yutaka, Venturi, Giulia, Rebholz-Schuhmann, Dietrich, and Ananiadou, Sophia. (2011). The biolexicon: a large-scale terminological resource for biomedical text mining. *BMC Bioinformatics*, 12(1):397.

UNI-10653. (2003). *Qualitá della documentazione tecnica di prodotto*. UNI.

Venturi, Giulia, Montemagni, Simonetta, Marchi, Simone, Sasaki, Yutaka, Thompson, Paul, McNaught, John, and Ananiadou, Sophia. (2009). Bootstrapping a verb lexicon for biomedical information extraction. In Gelbukh, Alexander, editor, *Computational Linguistics and Intelligent Text Processing*, volume 5449 of *Lecture Notes in Computer Science*, pages 137–148. Springer Berlin Heidelberg.