

# Ambiguity Management in Natural Language Generation

Francis Chantree

The Open University

Dept. of Maths & Computing,

Walton Hall, Milton Keynes, Buckinghamshire, MK7 6AA, U.K.

Email: [f.j.chantree@open.ac.uk](mailto:f.j.chantree@open.ac.uk)

## Abstract

In natural language generation (NLG) it can be desirable to preserve some ambiguities rather than try to eliminate them. The question is, which ones are so liable to misinterpretation that they should be removed, and which can be allowed to remain. Our approach to the problem is to notify the user of an NLG system of ambiguities that might be present in generated text, and to advise him/her as to how serious these might be for readers of the text. The degree to which each ambiguity is considered serious is adjusted by the responses of users upon seeing instances of that ambiguity. The system learns from these interactions, and an ambiguity tolerance level is thereby established. The ultimate goal of this research is to build an adaptive ambiguity notification tool that incorporates these features and can be used with an NLG system.

## 1. Introduction

We are studying the problem of ambiguity in (NLG) and, in particular, the ways in which it can be managed in the process of generating text. Managing ambiguity takes the form of either eliminating it or preserving it. While the former is the most obvious approach, it is not always feasible or desirable. Some ambiguities can safely be left in the text. The problem is, which ones can be left and which ones must be removed. Our solution to this is an ambiguity notification tool which informs the user of the NLG system about potential ambiguities in the text. It does this at present by the use of part of speech triggers obtained by tagging the text. The user responds by saying how serious the ambiguity is to readers of the text. The system learns from the users' responses and creates a level of seriousness for each type of ambiguity based on these. It is thereby adaptive to the users of the system. A level of seriousness, below which an ambiguity will be

tolerated, forms the ambiguity tolerance level for that ambiguity. Using the adaptive process, the ambiguity tolerance level can be adjusted to the needs of the user, to the domain or to a company's requirements. A base level of seriousness for each ambiguity is set prior to the adaptive process.

We are using requirements specifications as our domain. This has been chosen this as it is a field where ambiguity must be kept to a minimum due to the high cost incurred should inaccurate specifications be converted into faulty equipment or systems. Authors writing requirements specifications should be more prepared than most to spend extra time on the process of disambiguation and, we hope, on learning how to use our tool.

It is intended that the tool will be used with a specific NLG system, WYSIWYM (Power et al. 1998). This system is suitable because it provides for directly manipulation of the knowledge base, and because it is already designed to give feedback to the user.

## 2. Ambiguity Management

### 2.1 The Basics of Ambiguity

The basic definition of ambiguity, as generally used in natural language processing, is "capable of being understood in more than one way". It can be classified into many different types and using various different classification schemes. The most widely used classification is probably the one which divides ambiguity into *lexical ambiguity*, whereby a word may have more than one interpretation; *semantic ambiguity*, whereby several interpretations result from the different ways in which the meanings of words in a phrase can be combined; *syntactic ambiguity*, whereby several different interpretations result from the different ways in which a sequence of words can be grammatically structured; and *pragmatic ambiguity* whereby the context of a phrase results in there being alternative interpretations of that phrase (Kamsties 2001). There are also other linguistic concepts which are akin to

ambiguity and which may also lead to misinterpretations. These concepts include *generality*, whereby one meaning subsumes another meaning and yet the same word is used for both; *indeterminacy*, whereby there are factors which, though they are not necessary for understanding a meaning, are necessary for establishing what it refers to; and *vagueness*, whereby one is always uncertain of a meaning regardless of how much one tries to qualify it.

Some words and linguistic constructions are said to be *inherently* ambiguous - or at least "vague" (Kamsties 2001). For example, in the phrase "This process is rapid", it is unclear whether the adjective "rapid" is being used in relation to the other processes contained within that system or perhaps to similar processes in rival systems. It is likely that the NLG system would wish to either avoid such vague words or to qualify them as much as possible.

Ambiguity is an ever-present feature of language, and people perceive ambiguities differently. This is seen when text is interpreted by a variety of people who have their own set patterns of interpretation. For example, with those whose first language is not English, idioms become problematic linguistic constructions, and there is always a danger of an interpretation that is too literal or of an interpretation that chooses an unlikely meaning of a word.

## 2.2 Ambiguity in NLG Systems

Although complete disambiguation is not possible, designers of NLG systems try to avoid most ambiguities. They may then choose to take the view that the user will correctly understand what is meant, even if he/she doesn't actually understand the intricacies of disambiguation (Hobbs 1983). But, as NLG systems are designed to generate more complicated surface realisations than previously, ambiguity becomes more of a problem. This is very often due to the desire to generate everyday formulations of language. For example, in the sentence "Open the case with the key", the key is probably a tool which should be used to open the case rather than it being a feature of the case. The decision of whether to disambiguate this sentence or not might depend upon the users' proficiency with English and the context provided by the surrounding text. (The ambiguity here is due to the presence of the *preposition* "with"; it is an example of *attachment ambiguity*, which is a syntactic ambiguity.)

Conversely, an ambiguity can be desirable due to the same desire to write concise easily understood text, see e.g. (Shemtov 1997). For example, in the instruction "Take two tablets twice a day with a glass of

water", we know what is meant. Yet, in theory, it could mean that the same glass of water must be used for all the tablets that are taken (Kibble, Power & van Deemter 1999). (The ambiguity here is due to the presence of *quantifiers* and it is an example of *scope ambiguity*, which is a semantic ambiguity.) We may not want to elaborate the instructions any further as it is commonly understood that the quantity of water consumed is not important. Users are accustomed to resolving many types of ambiguity in texts subconsciously and efficiently using common-sense knowledge. However, the computer cannot be supplied with the wealth of common-sense knowledge that is necessary to resolve all ambiguities. Therefore, it does not know which of the surface realisations that it generates will be disambiguated correctly by the user - or left ambiguous because the consequences of misinterpretation are trivial - and which will be genuinely problematic. Also, even if the generator is able to find alternative wordings for ambiguous phrases, this can lead to tortuous and convoluted text and can therefore be counter-productive. Consequently, resolution of ambiguity is not always the answer and it can be very beneficial to allow some ambiguities to remain in the text. This process is known as "ambiguity preserving".

## 3. Previous Work

### 3.1 Minimising Ambiguity

Designers of natural language generation systems seek to minimise ambiguity implicitly by avoiding the use of words and linguistic constructions that might introduce ambiguity into the text. This is brought about by use, either explicitly or implicitly, of a controlled language (CL). To illustrate this, we discuss a couple of explicit uses of CLs here. (Danlos, Lapalme & Lux 2000) introduce the notion of an Extended Modular Controlled Language (EMCL) which is designed with NLG in mind. It is modular in that, while some syntactic rules are seen to be universally desirable and are therefore applicable to the documentation task as a whole, others have more localised relevance and are therefore enforced only in sub-modules that cover a part of the documentation task. This is akin to our notion of a flexible approach to ambiguity whereby the acceptance of particular ambiguities depends on context and the wishes of the user. (Power, Scott & Hartley 2003) take these ideas a step further by introducing the idea that a controlled language can actually control the *meaning* of what is expressed as well as the structure. They are able to do this by using a system, WYSIWYM, which allows direct manipulation

of semantics (Power et al. 1998). Both sets of researchers claim that their approaches hold great possibilities, for multilingual generation in particular. Multilingual generation is one of the chief motivating factors behind NLG research, and this makes WYSIWYM a promising system for us to be working with.

### 3.2 Preserving Ambiguity

For the reasons mentioned earlier, some researchers have proposed methods for preserving ambiguity, e.g. (Shemtov 1997), (Emele & Dorna 1998) and (Knight & Langkilde 2000). Our objectives are most closely in line with the last of these: we cover a wide range of ambiguities and there is a statistical element implicit in our approach. The statistical element of (Knight & Langkilde 2000) involves "thinning out" the large number of candidate parse trees produced by their generator using statistical analysis of suitable corpora. They claim that their technique is highly effective, and simpler than those of the other researchers mentioned whose algorithms are more complicated and inflexible. However, their statistical analysis is reliant on the corpora that they scan. It is unlikely that such corpora would contain many instances of the specialised technical language that occur in requirements specifications documents. Their system could indeed be trained on requirements specifications documents, but these vary so greatly that many instances of specialised technical language would probably not be sufficiently represented. Also, their statistical analysis is not able to take into account the large, and variable, amount of context that is useful for sophisticated management of ambiguity. We believe, therefore, that it is better to utilise the user's own understanding directly and effectively in the management of ambiguity, in a way that is adaptive to the task at hand.

### 3.3 Interactive Disambiguation

The idea of interactive disambiguation of text has been considered before, for instance by researchers working on the KANT machine translation project (Mitamura 1999). KANT's on-line authoring system supports interactive disambiguation of lexical and structural ambiguities. If a sentence is considered to be ambiguous in the grammar checking phase, the system is able to indicate whether a lexical ambiguity or a structural ambiguity is the cause. It would seem that the only structural ambiguity covered is attachment ambiguity. The author is asked to choose the intended meaning for the word in question or the intended point of attachment for the preposition. Alternatively, he or

she may choose to rewrite the sentence as an unambiguous surface realisation.

### 3.4 Weighting Ambiguity

Louisa Mich - sometimes working in association with Roberto Garigliano - is also concerned with ambiguity in the domain of requirements engineering, see e.g. (Mich 2001). She concentrates solely on two levels ambiguity, semantic (including lexical) and syntactic, and ignores all notion of context. Her treatment of semantic ambiguity is simply a function of the number of possible meanings of words. She then gives a formula for "weighted semantic ambiguity" that incorporates the frequencies that the different meanings of a word occur in. Her treatment of syntactic ambiguity is a function of the number of possible syntactic roles that a word can take. She then gives a formula for "weighted syntactic ambiguity" that incorporates the frequencies that the different roles of a word occur in. Mich has tested her formulae on the menu commands of an Internet browser, and has used two different lexical resources for evaluating semantic and syntactic ambiguity.

Ambiguities can also be quantified by more empirical methods. Researchers working with the KANT machine translation system made some basic quantifications of the efficacy of the disambiguation methods that they had proposed (Baker et al 1994). The factors that they analysed were the introduction of a constrained lexicon, a constrained grammar, and semantic restriction using a domain model, and the limiting of noun-noun compounding. Constraining the lexicon produced far and away the best results, followed by restriction of the grammar and then by limiting of noun-noun compounding. Combinations of these factors produced cumulative benefits.

## 4. Approaching a Solution

### 4.1 Classification of Ambiguities

For reasons of completeness, we choose to consider all causes of possible misinterpretation in our initial remit. Although our work is currently using only part of speech triggers contained within the text, there are in fact many types of ambiguity that this approach can warn about. Lexical and syntactic ambiguities are the most obvious candidates, but some types of semantic ambiguity and pragmatic ambiguity are also amenable to our approach, as is discussed briefly later. Based the work of (Kamsties 2001), we also take consideration of the ambiguities that are specific to our chosen domain, requirements engineering.

All the forms of ambiguity that we have considered

are then arranged in a classification that is suited to our purpose. This classification is then adjusted by taking into account the results of our own corpus analysis, and expert analysis of the appreciation of ambiguities in the domain. The corpus analysis reveals the prevalence of the various types of ambiguity but not their seriousness. The expert analysis gives an indication of the seriousness of the various types of ambiguity but not their prevalence. These two methods of adjustment, which are described below, are combined to give an amended and weighted classification of ambiguities. This is used to derive an initial level of seriousness for the each type of ambiguity. Because the corpus analysis and the expert analysis take averages of all the results that are elicited, this represents the "implicit statistical alluded" aspect alluded to earlier.

#### 4.2 Corpus Analysis

We have collected a corpus of requirements specifications documents, and have tagged them with part of speech tags using a trained Brill tagger (Brill 1992). We have found these documents to be quite variable in the quality of grammar and sentence construction used. This has both good and bad implications: the ambiguities that we have found should be picked up our system, but instances of bad English will not be corrected and may in fact cause our system to perform incorrectly. Our corpus analysis takes the form of scanning the documents for actual ambiguities and then noting these ambiguities in the texts. This process of ambiguity detection is achieved by use of a Perl program that scans the tagged text for combinations of parts of speech that indicate the possible presence of ambiguities. The presence of these combinations is a simple way of beginning the process of ambiguity detection. For instance, the presence of prepositions and multiple conjunctions indicates possible *attachment ambiguities* and *coordination ambiguities* respectively. Also, quantifiers - such as "all" and "a" - are notorious as indicators of *scope ambiguity*. The Perl program, as well as locating these lexical triggers, also locates the phrases surrounding them which will be potentially affected by the ambiguity. So, for example, multiple noun phrases that occur immediately before a preposition are stored as alternative candidates to which the preposition may be attached. This reinforces the possibility of a potential attachment ambiguity, and it demonstrates to the user where the ambiguity might lie. We aim to increase the sophistication of this procedure so that a wider range of possible ambiguities can be detected and so that fewer situations where ambiguity is not a problem will be flagged.

#### 4.3 Analysis by Experts

In the second stage of adjusting our classification of ambiguities, a group of requirements engineering experts are shown representative examples of the ambiguities discovered in the corpus analysis. They are then asked to evaluate to what extent the linguistic constructions presented to them might be misinterpreted. They are given the context of the ambiguity similar to the context that was represented in the text but without any application-specific language which they would not be expected to know. Because of the differing amounts of relevant context that occur with different linguistic constructions, the experts are presented with potential ambiguities that are often quite similar but that have contexts of different informativeness. The evaluations are used to create weightings that are applied to the types of ambiguities in the classification. These weightings are the initial weightings for the interactive ambiguity notification tool. These are used by the tool before it is put into use by users who change the tolerance level, and the associated seriousness levels.

The idea of weighting ambiguities described in (Mich 2001) is not rigorous enough in our opinion. No account is taken of context, which leaves a great amount of potential ambiguity - scope ambiguity and referential ambiguity, for example - unconsidered. Simply measuring the diversity of meanings and roles that a word can have misses many of the major sources of ambiguity which are not seen at a lexical level.

### 5. The Ambiguity Notification Tool

#### 5.1 Introduction

We are creating a computerised system that will achieve the two key goals of our research: detecting possible ambiguities, and advising the user as fully as possible of their significance. This is the "ambiguity notification tool" which will provide feedback to users of a natural language generation system. This feedback will give notification of possible ambiguities in generated texts, of their estimated seriousness, and of options to change them. The user will create the texts that are to be generated, as described in the NLG Tool section below, and the feedback relating to the potential ambiguities will also be displayed. With each potential ambiguity the user will then have the opportunity to alter the texts, using WYSIWYM's capabilities, whereby the system will automatically be notified that that ambiguity was genuinely problematic. Alternatively, the user can choose to let the ambiguity remain in the text. A similar idea is used in the KANT

machine translation system (Baker et al. 1994). After the system has carried out as much disambiguation as possible, any remaining lexical and structural ambiguities are described to the user. He or she may then select the desired interpretation from the possible surface realisations supplied by the system. This indicates that user participation can be of value even with sophisticated and mature language processing systems. The ambiguity notification tool has not been built yet so this section represents a plan for future work.

## 5.2 Seriousness and the Tolerance Level

Initially, base-line levels for the seriousness of the different ambiguities are set using the adjusted and weighted classification of ambiguities. Thereafter, the seriousness levels are adjusted in accord with the responses of users to the ambiguities that they are presented with by the system. The system "learns" from these interactions, and a tolerance level is thereby established. The tolerance level is a level of seriousness below which an ambiguity is tolerated. Ambiguities whose level of seriousness is below the tolerance level can be annotated in a different way to more serious ambiguities, or they can simply be allowed to remain in the text unannotated. Also, ambiguities at other levels of seriousness can be annotated in different ways. For instance, very serious ambiguities can be distinguished from less serious ones, even though both types are above the tolerance level, so that priority can be given to resolving of the former types. Certain words and linguistic constructions will become associated with serious ambiguities and others will become associated with trivial ones. The tolerance level and seriousness levels can be reset at any point.

## 5.3 Adaptivity

The system will be adaptive to the user and, if required, to the domain that is being used. The tolerance level that has been established by a user can be applied whenever that user has further interactions with the system, or when the system is used by other users who are working within the same domain. Also, a company might wish to apply one user's tolerance level to all its other users should that first user be deemed to be an author of paradigmatic prowess. The system will therefore be trained to advise the user, based on previous exposure to similar texts, as to the best decision at each instance of a potential ambiguity.

The adaptive aspect of our work takes its cue from research on adaptive information extraction for the Semantic Web (Ciravegna & Wilks 2003). These

researchers have highlighted the need for such a system to be able to adapt to aspects of language that are specific to distinct domains, applications and genres. These aspects of language include grammars, lexica and discourse structures. Their system learns information extraction rules from a specific training corpus annotated with XML tags, and can then apply these rules to similar corpora. The researchers claim that their system is a useful tool, especially for naive users, even with fairly limited training. Similarly, we hope to help users who may not be entirely at home with the English language by providing a tool which is suited to their authoring task and has been trained up by experts.

## 6. The NLG Tool

The natural language generation system that we intend to use is WYSIWYM (Power et al. 1998). This system is suitable because it offers directly manipulation of the knowledge base, and because it is designed to give feedback to the user. The acronym stands for What You See Is What You Meant: what you see being feedback texts displayed to the users, which show them what they meant when they chose linguistic data from the knowledge base to form sentence constructions. WYSIWYM has been recently re-arranged so that it has a (more advantageous) semantically-orientated knowledge base (Power et al. 2003). The domains in which it has been used include patient information leaflets and maritime rules.

The direct manipulation aspect of WYSIWYM is important as we want the users to be able to change texts immediately if they encounter ambiguities that they consider to be serious. It is also useful to us that WYSIWYM is already producing feedback texts that give an idea about what the surface realisation of the data structures that have been built up by the user will look like. It should be comparatively straightforward to adapt this capability to produce feedback texts that warn about ambiguity.

## Acknowledgments

Thanks are due to my supervisors Anne de Roeck and Bashar Nuseibeh, and to Alistair Willis.

## References

- Baker, K. L., Franz, A. M., Jordan, P. W., Mitamura, T. & Nyberg, E. H. 1994. "Coping With Ambiguity in a Large-Scale Machine Translation System". In *Proceedings of COLING-94, vol.1*, Tokyo, Japan.
- Brill, E. 1992. A Simple rule-based part of speech

- tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Somerset, NJ, U.S.A.
- Ciravegna, F. & Wilks, Y. 2003. "Designing Adaptive Information Extraction for the Semantic Web in Amilcare". To Appear. (University of Sheffield, U.K.)
- Danlos, L., Lapalme, G. & Lux, V. 2000. "Generating a Controlled Language". In *Proceedings of the First International Conference on Natural Language Generation (INLG 2000)*, Mitzpe Ramon, Israel.
- Emele, M. C. & Dorna, M. 1998. "Ambiguity Preserving Machine Translation using Packed Representations". In *Proceedings of the 17th International Conference on Computational Linguistics (COLING-ACL '98)*, Montréal, Canada.
- Hobbs, J. R. 1983. "An Improper Treatment Of Quantification In Ordinary English". In *Proceedings of the Twenty-First Annual Meeting of the Association for Computational Linguistics*, Cambridge MA, U.S.A.
- Kamsties, E. 2001. "Surfacing Ambiguity in Natural Language Requirements". PhD Thesis, Fraunhofer-Institut für Experimentelles Software Engineering, Kaiserslautern, Germany.
- Kibble, R., Power, R. & van Deemter, K. (1999) Editing logically complex discourse meanings. In *Proceedings of the 3rd International Workshop on Computational Semantics*, Tilburg, The Netherlands.
- Knight, K. & Langkilde, I. 2000. "Preserving Ambiguities in Generation via Automata Insertion". In *Proceedings of Twelfth Conference on Innovative Applications of Artificial Intelligence*, American Association for Artificial Intelligence, Austin, Texas, USA.
- Mich, L. 2001. "Ambiguity Identification and Resolution in Software Development: a Linguistic Approach to Improving the Quality of Systems". In *Proceedings of Seventh IEEE Workshop on Empirical Studies of Software Maintenance*, Florence, Italy.
- Mitamura, T. 1999. "Controlled Language for Multilingual Machine Translation". In *Proceedings of Machine Translation Summit VII*, Singapore.
- Power, R., Scott, D. & Evans, R. 1998. "What You See Is What You Meant: direct knowledge editing with natural language feedback". In *Proceedings of the 13th Biennial European Conference on Artificial Intelligence*, Brighton, U.K.
- Power, R., Scott, D. & Hartley, A. 2003. "Multilingual Generation of Controlled Languages". In *Proceedings of the 4th Controlled Language Applications Workshop (CLAW03)*, Dublin, Ireland.
- Shemtov, H. 1997. "Ambiguity Management in Natural Language Generation". PhD Thesis, Stanford University, U.S.A.