GoalGetter: Football results, from teletext to speech

E.A.M. Klabbers[1], J.E.J.M. Odijk, J.R. de Pijper and M. Theune[1]
e-mail: klabbers@ipo.tue.nl

Abstract

This paper shows that a new Data-to-Speech system, called GoalGetter, could be created relatively easily from an existing one, the DYD system, despite the fact that the domain, the language and the speech output technique differ. It is shown that the original system is easily portable and that problems that arose after porting can receive generalized solutions so that the system as a whole is enhanced. The speech output technique used (a specific variant of the phrase concatenation technique) finds an optimum between the requirements of high quality output and flexibility. In addition, requirements for setting up a phrase database necessary for this technique are specified.

Introduction

This paper describes the GoalGetter system, which generates spoken reports of football matches. The input of the system is a text in table format and a small database. The text is stored on a Teletext page and contains data on one or more football matches. The database contains information about the teams and their players. The output of the system is a correctly pronounced, coherent text in Dutch which conveys the information on one of the matches of the Teletext page.

We concentrate on two modules of the GoalGetter system: the text generation module and the speech output module. The construction of the text generation module was accomplished by porting large parts of an existing module of another text generation system, the DYD system (see Van Deemter, Landsbergen, Leermakers & Odijk, 1994; Odijk, 1995; Van Deemter & Odijk,1995), to an application with a different language and a new domain. The speech output module of the GoalGetter system uses a version of the phrase concatenation technique, in which prosodically different variants of phrases are correctly combined.

GoalGetter might be part of a more general automatic information service application, for example a telephone service. The system could also be used in situations where eyes and hands are occupied (e.g. in a car), or in combination with other modes (e.g. textual or graphic modes). A main goal with making this system was to acquire knowledge about and experience with the construction of Data-to-Speech systems for different domains and languages, using various text generation and speech output techniques. The language generation technique used in GoalGetter was previously used in the DYD system. The DYD system contains a Data-to-Speech system in the domain of music. It generates text in English and uses formant synthesis for speech output. In the near future we intend to adapt the same technique for use in a large travel information system (the OVIS-

1. Authors Klabbers and Theune carried out this research within the framework of the Priority Programme Language and Speech Technology (TST). The TST-programme is sponsored by NWO (Netherlands Organization for Scientific Research).

system[1]) and in other applications.

A system which is related to the GoalGetter system is the SOCCER system (see André, Herzog & Rist,1988; Herzog & Retz-Schmidt, 1989). The SOCCER system generates spoken natural language descriptions of (events in) football matches, based on and simultaneously with analyses of image sequences of football scenes. The GoalGetter system, in contrast, takes tabular information of the course and the result of a match as input. This is comparable to the sports summaries generated by the STREAK system described in Robin (1994) and McKeown, Robin and Kukich (1995). The sports domain in the STREAK system, however, is basketball, which has consequences for the character of the texts generated. In addition, STREAK does not produce spoken output.

The GoalGetter system consists of three modules: (1) a preprocessor to convert Teletext pages into a format which is suitable for the text generation module; (2) the text generation module (TGM) which takes as input these data and data from a small database with information about the teams and their players, and yields as output prosodically annotated text, called *enriched text*; (3) a speech module, which takes as input enriched text, and yields as output a speech signal which ideally mimics the natural pronunciation of the enriched text.

First, we will briefly describe the general architecture of the TGM. Next we describe our experiences with porting this module from the DYD system to the GoalGetter system, some specific problems encountered in the texts generated, and proposals for generalizing the TGM to avoid such problems. After that we discuss the technique of phrase concatenation and how the relevant speech database should be constructed. We end with some concluding remarks.

General architecture

The general architecture of the TGM is depicted in Figure 1. It consists of two modules, *Generation* and *Prosody*, and three data resources: 1) a set of *templates*; 2) the *Knowledge State*; and 3) the *Context State.* The data concerning the results of a particular football match and the data on the teams and their players (the *domain data*) are one part of the input for *Generation*. *Generation* also uses a collection of *syntactic templates* internal to the system. Syntactic templates, informally, are syntactic parse trees for sentences or sentence parts, with slots for variable parts, and with conditions governing their use. Some of these conditions are formulated as conditions on the *Knowledge State*. The *Knowledge State* records which data have been conveyed, and which have not yet been conveyed. The *Generation* module checks whether the conditions associated with syntactic templates evaluate to true, and in this way determines in this manner which syntactic templates can be used at the current point in the text. These conditions must ensure that a coherent text results (for details, see Odijk, 1995).

If a syntactic template can be used, one or more syntactic trees for sentences can be generated from it by filling its slots with syntactic trees generated from other syntactic templates. Sentence-internal syntactic conditions determine which of the resulting syntactic trees, if any, are well-formed. For each syntactic structure generated, the *Generation* module finds out whether it is appropriate at the current point in the text by checking con-

1. Information on OVIS can be found at: http://grid.let.rug.nl:4321

ditions formulated on the *Context State*. These conditions concern the use of referential and quantificational expressions. If more than one syntactic structure satisfies all these conditions, one is selected arbitrarily and used in the text. *Generation* updates the *Knowledge State* and the *Context State* accordingly.

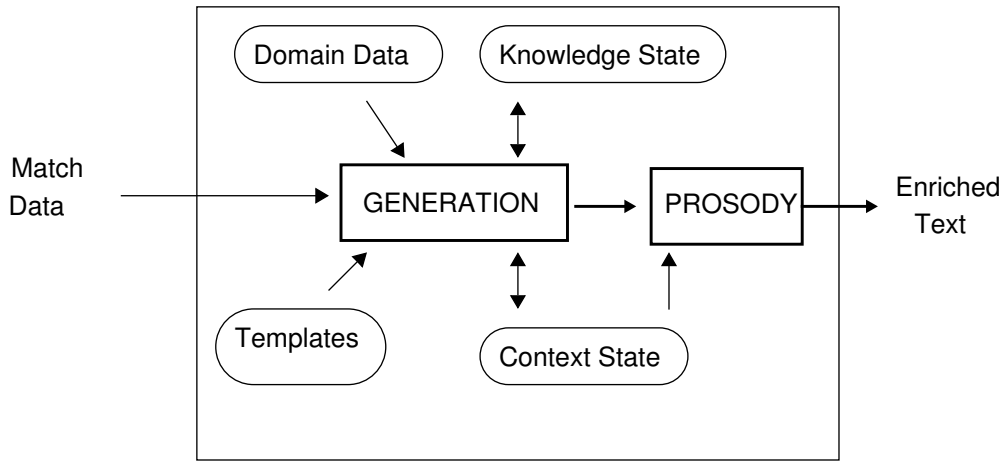

Figure 1: The architecture of the text generation module

The output of *Generation* is a syntactic tree for a sentence which is input to the *Prosody* module. The *Prosody* module converts the syntactic tree into a metrical tree and assigns accents and prosodic boundaries taking the properties of the *Context State* into account. For example, 'given' information is deaccented (see Hirschberg, 1992; Van Deemter, 1994). In this manner the prosodic properties of a sentence are co-determined by the preceding discourse. A metrical tree is binary-branching and each pair of branches has a strong and a weak branch. The determination of strong and weak branches and the assignment of accents and prosodic phrase boundaries is carried out by an algorithm based on a version of Focus-Accent Theory (Dirksen,1992; Dirksen & Quené, 1993). The output of the *Prosody* module is a sequence of words with prosodic annotations (an *enriched text*). For more details about the text generation module we refer to Van Deemter et al. (1994), Collier and Landsbergen (1995), Odijk (1995), Van Deemter and Odijk (1995) and Van Deemter, Van der Hoeven, Leermakers, Odijk and Uittenbogaard(1996).

Portability of the text generation module

In this section we discuss our experiences in porting the text generation module from the DYD system to the GoalGetter system. An overview is given of the actions that are necessary to port the module to a new domain, a new language and a new speech generation method respectively.

A new domain

Porting the TGM to a new domain requires the creation of a whole new set of templates. This is not a difficult task, but it is time-consuming, even though in many cases parts of

'old' templates can be recycled for use in the new domain. This drawback is inherent to all template-based generation systems.

The other parts of the TGM - the *Context State*, the *Knowledge State* and the *Prosody* module, as well as the general *Generation* component - can remain virtually unchanged. We only need to specify which parts of the new input data structure require to be expressed, and which are the possible topics of the new templates.

A new language

If the language[1] of the application changes, and the domain remains the same, in most cases only the syntactic information in the existing templates has to be modified.

In addition, some adjustments to the *Prosody* module are needed. The deaccentuation rules in this module are assumed to be language-independent, but they make use of a list of unaccentable words, e.g., function words, which has to be replaced with a list of words for the new language. Accentuation and placement of phrase boundaries are determined by means of language-independent parameterized rules. Since the relevant parameters may differ between languages, they may have to be modified when the system is ported to a new language. The rules remain unchanged.

All other linguistic rules of the TGM, e.g., conditions governing referring expressions, are assumed to be language-independent as well. So, porting to a new language mainly requires partially rewriting the templates and specifying new parameters for some rules.

A new speech output technique

The TGM can be used with different methods of speech output. Because the text to be spoken is generated by the system itself, it is fairly easy to determine its prosodic features (especially in comparison to text-to-speech systems) on the basis of the information from the TGM.

Conclusion

The effort needed to port the TGM to another domain, language or speech output technique is low enough to make it a worthwhile enterprise. Moreover, it can be done by linguists with little background in computer science. Porting the TGM mainly involves the rewriting of templates. The other components are mostly domain- and language-independent and require only minor changes.

Referring expressions

As we explained in the preceding section, the majority of rules used in the TGM are domain-independent. However, when porting the TGM from DYD to GoalGetter we found out that a few rules concerning referring expressions were insufficiently general. Our aim is to find general formulations of these rules which make them domain-independent.The first rule we discuss concerns the use of proper names, and the second one concerns accentuation of referring expressions.

The distance between proper names

1. We restrict ourselves to the class of related European languages like English, Dutch, German, etc.

Conditions governing the use of referring expressions are checked on the *Context State*. Some of those conditions are stylistic in nature: violating them does not cause any real ungrammaticality, but results in stylistically 'ugly' texts. An example is the Distance Condition for proper names, which states that identical proper names with the same reference should not occur too close together, i.e., in the same paragraph.

In the football domain, however, repetition of the same proper name cannot be avoided. With each football event described, we need to identify the player and the team. It is often necessary to use a proper name to refer to them, because it is impossible to use a pronoun or a definite description. However, if the proper name was used earlier in the same paragraph, the Distance Condition forbids it and makes reference impossible. This problem will occur in every domain where referring expressions must be used regularly.

The following solution is proposed. First, we will implement the Relaxed Distance Condition for proper names. This is a less strict version of the Distance Condition, which only forbids the repeated use of the same proper name in one paragraph when a pronoun or definite description could be used for reference as well.

Second, we will implement a Distance Condition governing definite descriptions which is similar to the original Distance Condition relating to proper names. It disallows repeated use of the same definite description in one paragraph, which is stylistically even worse than repetition of the same proper name. The reason for this may be that the main purpose of definite descriptions is to provide extra information about the object they identify (cf. Maes, 1991).

For instance, the definite description of Ajax as *de ploeg uit Amsterdam* ('the team from Amsterdam') provides additional information about the team's city of origin. If this description is used for a second time, however, it no longer conveys any extra information, because this information has already been communicated. The definite description no longer fulfils its main purpose, and therefore seems out of place.This is in line with the observation in Grosz, Joshi and Weinstein. (1995, p. 216), that the realization of a backward looking centre in Centering Theory as a 'full definite noun phrase' is best when the noun phrase conveys some additional information about the object it refers to.

Following Maes (1991), who states that literally repeated proper names have an identificational function only, we assume that proper names are purely used for reference, not for conveying extra information. This means that their use is less restricted. So, if there is a choice between repeating the same definite description or the same proper name, the latter is to be preferred. This is achieved by the combination of the Distance Condition governing definite descriptions and the Relaxed Distance Condition governing proper names, which ensures that we can always use a proper name for reference if neither a pronoun, nor a definite description can be used (e.g., because there is no additional information available).

(De)accentuation of referring expressions

The second rule we found insufficiently general concerns accentuation of referring expressions. As was said earlier, the *Prosody* module deaccents 'given' information. In our current implementation, an expression is regarded as given if it has the same reference as a preceding expression in the same paragraph.[1] Paragraphs in the output texts correspond

1. There are some additional causes for givenness, which are not relevant for the current discussion.

to topics in the discourse. We assume that a topic shift causes items to lose their 'givenness', as in Hirschberg (1992).

The texts generated in GoalGetter contain too many cases of incorrect deaccentuation, as in the following example, where the second occurrence of the name *Koeman* is wrongly deaccented because it is regarded as given:

(1) a Feyenoord nam na twee minuten de leiding door een goal van Koeman.
'Feyenoord took the lead after two minutes through a goal by Koeman.'
b Na vierentwintig minuten scoorde Witschge voor Feyenoord.
'After twenty-four minutes Witschge scored for Feyenoord.'
c In de drieënvijftigste minuut scoorde Koeman opnieuw.
'In the fifty-third minute Koeman scored again.'

Examples like this might suggest that our definition of givenness is insufficiently restrictive. However, following Prevost (1995), we will assume that it is not an incorrect notion of givenness which causes the problem, but our lack of a notion of *contrast accent*. In virtually all cases of incorrect deaccentuation the given noun phrase appears to be contrasted with a piece of information from the preceding sentence. This suggests that our accentuation problems would be solved if we had a rule for the assignment of contrast accent.

For the implementation of such a rule we must have a theory that tells us which items receive contrast accent in which circumstances. This theory should be based on some notion of semantic parallelism, because neither syntactic parallelism nor contrariety[1] can fully account for the cases of contrast we typically encounter in the football domain. Explanations in terms of 'sets of alternatives' (for instance Rooth, 1992; Prevost, 1995) are not sufficient because the presence of a member of the alternative set of an item does not always trigger contrast accent:

(2) a In de zesentwintigste minuut scoorde Koeman voor Feyenoord.
'In the twenty-sixth minute Koeman scored for Feyenoord.'
b Hierdoor kreeg Ajax een achterstand.
'This caused Ajax to be one goal down.'
c Twintig minuten later maakte Trustfull een doelpunt voor Feyenoord.
'Twenty minutes later Trustfull scored a goal for Feyenoord.'

An 'alternative set' theory of contrast would predict that *Feyenoord* in (2)c should have contrast accent, due to the presence of *Ajax* in (2)b. In fact, though, *Feyenoord* should be deaccented. This suggests that contrastable items like *Ajax* and *Feyenoord* do not receive contrast accent in all circumstances. We conjecture that they only have contrast accent when they occur in sentences that show a certain degree of semantic parallelism. Sentences (2)b and (2)c lack this parallelism; (2)b describes the current score in the match, while (2)c is a description of a goal-scoring event. If we replace (2)b by a sentence which describes a goal as well, e.g., *Na acht minuten herstelde Ajax het evenwicht door een doelpunt van Kluivert* ('After eight minutes, Ajax restored the balance through a goal by Kluivert'), both sentences are semantically parallel and *Feyenoord* does receive contrast

1. For a theory of contrast in terms of syntactic parallelism and contrariety, see Van Deemter (1995).

accent.

What we need is a formalized notion of semantic parallelism, and a specification of how much parallelism between sentences is required to trigger contrast accent, and on which parts of contrasted sentences the accent should land. With regard to the last question, the 'alternative set' theories might come in useful for establishing the contrasting items in a sentence pair.

Although our ideas about contrast accent are still very tentative, it is clear that the implementation of a rule for its assignment will greatly improve the performance of the system with respect to (de)accentuation. Although contrast accent did not play a prominent role in the music domain of DYD, in the football domain the lack of it caused severe problems, and we can expect this to be the case in other domains as well.[1]

To conclude, we can say that the addition of contrast accent and reformulation of the rules for proper names and definite descriptions, as suggested in the preceding subsection, will certainly increase the domain-independence of the Text Generation Module.

Speech generation

Introduction

There are various techniques that can be used to provide applications such as the GoalGetter system with speech output. One extreme is to simply record everything the system should be able to pronounce and play the recordings back. This approach gives great quality and little flexibility. The other extreme is to use unrestricted text-to-speech synthesis, which enables any text to be pronounced. This approach gives complete flexibility and inferior quality.

In GoalGetter, we have adopted a solution between these two extremes: entire phrases are prerecorded and played back in different orders to form complete utterances. In this way, a large number of utterances can be pronounced, based on a limited number of prerecorded phrases. Phrase concatenation to some extent reconciles the high fidelity quality and inherent naturalness of normal prerecorded speech with the flexibility of speech synthesis. This technique suits the GoalGetter application: a limited number of different utterances (carriers, templates) to be pronounced and variable information to be inserted in fixed positions (slots) in those utterances.

Phrase concatenation in GoalGetter

If all the necessary phrases are recorded in isolation and then simply concatenated, the resulting speech is likely to sound disjointed to the point where two people seem to be speaking at the same time. It is crucial to get the prosodic realization of the prerecorded phrases right, i.e., their loudness, rhythm and especially their pitch patterns. Therefore, in the GoalGetter system we use different prosodic variants for otherwise identical words and phrases to make the speech sound more fluent. Whenever a certain phrase is to be used in different contexts, it may have to be recorded multiple times, with different and carefully orchestrated prosodic realizations. To determine how many and what prosodic realizations should be recorded for each phrase, a thorough analysis of the material to be

1. If contrast accent is prosodically different from 'newness' accent, as was argued by Pierrehumbert & Hirschberg (1990) and Prevost (1995), this is an extra argument for the addition of contrast.

generated by the system is a necessary phase in the development of a phrase database.

Before the text which is generated by the application can be pronounced, we must determine which prosodic versions of the phrases available in the database are to be concatenated. In an information-providing system like GoalGetter, where text should be spoken in a neutral manner, the prosodic realization of phrases is largely determined by three factors: (a) whether one or more words in the phrase should be accented, (b) whether the phrase is followed by an important syntactic or prosodic boundary and (c) the strength of that boundary. These factors influence the parameters loudness, rhythm and pitch. In GoalGetter, the system generates the text and has reliable syntactic and semantic information available. Moreover, a fairly sophisticated algorithm is available in the TGM to assign accents and boundaries on the basis of this information. These accent and boundary markers are then used to trigger the choice of the appropriate prosodic variant from the phrase database.

Setting up a phrase database

Setting up a phrase database is a laborious task. Many factors influence the final result and must be taken into account both in the preparation stage and during post-processing, in order to obtain a good output quality. In the creation process a number of steps can be distinguished. Each of these steps is critical for the quality of the output. First of all, a recording script has to be constructed and a speaker selected. After that, recordings can be made and read into the computer for processing.

A *recording script* is a script for the speaker that lists all the utterances to be recorded. Before such a script can be constructed all the phrases and their prosodic variants must be known. In the case of the GoalGetter system the set of phrases and the phonetic contexts in which they occur have been deduced from the templates used to generate the texts.

The variable parts of sentences should not be recorded while spoken in isolation, but should be embedded in sentences that elicit the right prosodic variant from the speaker. When embedding the variable parts in running sentences, account must also be taken of the fact that in the postprocessing stage they have to be stored without their embedding sentences. Because the variable parts are cut out by visual inspection of the speech signal, they must have an identifiable start and end. Also, possible coarticulation between a variable part and the rest of the sentence should be kept to a minimum. To achieve this we have used fricatives most of the time in the immediately preceding and following contexts, since they are easily detectable.

The influence of the *speaker's voice* on the success of the application as a whole can hardly be overestimated. It is therefore very important to define strict requirements that the selected speaker must satisfy. The first choice is whether the speaker should be male or female. This depends on the type of application and the audience at which the application is aimed. Other requirements for GoalGetter were:

- The voice should be perceived as pleasant. This is perhaps the most important criterion, especially in a commercial application.
- The speaker should speak fluently.
- The speaker should not speak in a very flat and dull tone. It is easier to instruct a person who speaks too exuberantly to speak with less pitch variation than to convince a dull speaker to use more pitch variation.

- The speaker ought to have an ear for prosodic differences so that he can react properly to instructions about how to pronounce (intonate) a sentence.
- The speaker should not have a strong regional accent.

With the recording script ready and the speaker selected, *recordings* can start. Professional studio recordings are essential for a good output quality. In the future new material might have to be recorded, which makes extra demands on the recording conditions. Finally, the recorded phrases have to be cut out consistently, because in the concatenation process they have to link up perfectly.

Concluding remarks

The GoalGetter system generates spoken football reports in the Dutch language, based on tabular information on football matches. In our paper, we discussed the two main modules of the system: first, the text generation module (TGM) which produces natural language texts enriched with prosodic markers, and second, the speech output module which converts these enriched texts to speech.

Porting the TGM only requires a few minor modifications to its components, except for the templates. When the application domain or language changes, all templates have to be rewritten: a drawback which is inherent to all template-based generation systems.

In porting the TGM to a new domain, some of its rules turned out to be insufficiently general. We propose to reformulate the problematic rules in a manner which we believe to be domain-independent. Adding those new rules will make the system more robust and enhance its portability to new domains.

The fact that GoalGetter is a Data-to-Speech system, where the texts that have to be pronounced are generated by the system itself, makes it relatively easy to obtain the prosodic information which is needed for speech generation. In GoalGetter, we used a variant of the phrase concatenation technique where different prosodic versions of words and phrases were recorded. Which version of a word or phrase should be used, depends on its context and is indicated by prosodic markers in the output of the TGM. This technique produces a very natural-sounding speech output.

The text and the speech generation techniques employed in GoalGetter are both suited for use in many different limited domains. This, combined with the portability of the text generation module and the quality of the speech output, provides the basis for many possible future applications.

References

André, E., Herzog, G. & Rist, T. (1988). On the simultaneous interpretation of real world image sequences and their natural language description: The system SOCCER. *Proceedings ECAI 1988*, 449-454.

Collier, R.& Landsbergen, J. (1995). Language and speech generation. *Philips Journal of Research, 49(4)*, 419-437.

Deemter, K. van (1994). What's new? A semantic perspective on sentence accent. *Journal of Semantics, 11*, 1-31.

Deemter, K. van (1995). Contrastive stress, contrariety, and focus. Bosch, P. & van der Sandt, R. (Eds), *Focus & Natural Language Processing*, Cambridge University Press.

Deemter, K. van, Landsbergen, J., Leermakers, R., & Odijk, J. (1994). Generation of spoken monologues

by means of templates. *Proceedings of TWLT 8*, 87-96, Twente University.

Deemter, K. van & Odijk, J. (to appear). Context modeling and the generation of spoken discourse. IPO Manuscript 1125; Philips Research Manuscript NL-MS 18 728; to appear in *Speech Communication*.

Deemter, K. van, Hoeven, G. van der, Leermakers, R., Odijk, J. & Uittenbogaard, F. (1996). The use of natural language in a browsing interface. IPO Manuscript 1142.

Dirksen, A. (1992). Accenting and deaccenting: A declarative approach. *Proceedings of COLING 1992*, Nantes.

Dirksen, A. & Quené, H. (1993). Prosodic analysis: the next generation. Van Heuven & Pols (Eds), *Analysis and Synthesis of Speech: Strategic Research Towards High-Quality Text-to-Speech Generation*, Mouton de Gruyter, Berlin - New York.

Grosz, B., Joshi, A. & Weinstein, S. (1995). Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics, 21(2)*, 203-225.

Herzog, G. & Retz-Schmidt, G. (1989). Das System SOCCER: Simultane Interpretation und natürlichsprachliche Beschreibung zeitveränderlicher Szenen. Bericht 62, KI-Labor am Lehrstuhl für Informatik, Saarbrücken.

Hirschberg, J. (1992). Using discourse context to guide pitch accent decisions in synthetic speech. G. Bailly, C. Benoît and T.R. Sawallis (Eds), *Talking Machines: Theories, Models, and Designs*. Elsevier Science Publishers, Amsterdam.

Maes, A. (1991). *Nominal Anaphors and the Coherence of Discourse*. Ph.D. dissertation, Katholieke Universiteit Brabant.

McKeown, K., Robin, J. & Kukich, K. (1995). Generating concise natural language summaries. *Information Processing and Management, 31(5)*, 703-733.

Odijk, J. (1995). Generation of coherent monologues. Andernach, T., Moll, M. & Nijholt, A. (Eds.), *CLIN V: Proceedings of the Fifth CLIN Meeting*, 123-131, University of Twente.

Pierrehumbert, J. & Hirschberg, J. (1990). The meaning of intonational contours in the interpretation of discourse. Cohen, P.R., Morgan, J. & Pollack, M.E. (Eds.), *Intentions in Communication*, 271-311, MIT Press, Cambridge.

Prevost, S. (1995). *A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation*. Ph. D. dissertation, University of Pennsylvania.

Robin, J. (1994). Automatic generation and revision of natural language report summaries providing historical background. *Proceedings of the 11th Brazilian Symposium on Artificial Intelligence*. Fortaleza, CE, Brazil.

Rooth, M. (1992). A theory of focus interpretation. *Natural Language Semantics, 1*, 75-116.