Modeling the dative alternation with automatically extracted features*

Huayan Zhong and **Amanda Stent**
Computer Science Department
Stony Brook University
Stony Brook, NY 11794-4400
huayan, stent@cs.sunsyb.edu

Mary Swift
Computer Science Department
University of Rochester
Rochester, NY 14627
swift@cs.rochester.edu

Abstract

We show that generation of contextually appropriate syntactic variation can be improved using a model based on automatically extracted features. We adapt a model for predicting dative alternation from (Bresnan *et al.* 2005); this model incorporates lexical, syntactic, semantic and pragmatic features. We evaluate the effect of using different types of feature on this classification task and show that the most predictive features for text differ from those for dialog. Finally, we show that modeling this type of syntactic variation improves the performance of surface realization for both text and dialog.

Introduction

The ability to produce contextually appropriate syntactic variation contributes to the coherence and naturalness of natural language generation (NLG). Factors that contribute to such variation can be wide-ranging and complex and require models beyond corpus frequencies (e.g. (Creswell & Kaiser 2004)). In this paper we examine modeling one type of variation that affects the placement of arguments within the VP, the dative alternation. Dative alternation involves the variation between two different syntactic representations for the arguments of a ditransitive verb, i) the dative NP:

`Agent Verb Recipient Theme`
I gave the dog a bone

and ii) the dative PP:

`Agent Verb Theme [to Recipient]`
I gave a bone to the dog

In recent work, (Bresnan *et al.* 2005) describe a combination of lexical, syntactic, semantic and pragmatic features that model the dative alternation with high accuracy. For their experiments, they use hand-annotated features. In this paper, we show that it is possible to achieve comparable accuracy using automatically extracted training data and features. We compare the effect of different types of feature on this classification task, and show that the most predictive features for text differ from those for dialog. Finally, we show that modeling this type of syntactic variation improves the performance of statistical surface realization for both text and dialog.

The rest of this paper is structured as follows: first, we discuss previous approaches to predicting dative alternation, and give a brief description of research on surface realization. Second, we describe our data. Third, we describe our model for predicting dative alternation. Fourth, we describe our surface realization experiments. Finally, we summarize our results and describe future work.

Related Work

Predicting Dative Alternation

One analysis of the dative alternation assumes that the two structures have different semantic representations (e.g, (Pesetsky 1995; Harley 2000)). A contrasting approach (e.g. (Baker 1997)) argues that the two structures have the same semantic representation, and surface variation is motivated by the discourse theoretic status of the theme and recipient arguments. Previous research has explored the influence of factors such as animacy, definiteness and length of the arguments (syntactic weight) on the surface realization of the dative construction (e.g. (Collins 1995; Halliday 1970; Thompson 1990)). (Bresnan *et al.* 2005) use three different models predicting from multiple variables to explore the problem of what really drives the dative alternation. Model A is a logistic regression model using 14 features (features are described in Section) and shows that certain properties of recipient and theme (discourse accessibility, animacy, definiteness, pronominality and syntactic weight) are mutually irreducible. Model B is a multilevel logistic regression variant of Model A which shows that the features remain significant when conditioned on different verb senses. Model C is a variant of Model B adapted for available features in the data and shows that the features are predictive when tested on different corpora (Switchboard and Wall Street Journal). Bresnan et al. report the performance of the models: Model A, 92%; Model B, 94% and Model C, 93%. In this paper we

*We would like to thank the anonymous reviewers for their helpful comments. This work was supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010, via subcontract to SRI #03-000223).

explore the utility of such models for statistical NLG. We adapt Model A using automatically extracted data and features, evaluate its performance on data from both text and dialog corpora, and examine which features are most predictive. We then evaluate the impact of our model on surface realization of the dative alternation.

Surface Realization

Surface realizers generate text from an input representation that may be semantic or syntactic in nature. Traditionally, surface realizers were built using hand-written grammars or templates (e.g. (Elhadad & Robin 1997; Baptist & Seneff 2000; Channarukul 1999)). However, most recent work on surface realization has used a two-stage approach (e.g. (Langkilde 2002; Rambow & Bangalore 2000; Ratnaparkhi 2000)). The first stage uses a simple hand-written grammar that overgenerates (i.e. produces syntactically correct and incorrect outputs). The second stage uses a statistical language model to rank sentences output by the first stage. These surface realizers run quickly and produce high-quality output. However, the interaction between grammar and language model, and the lack of modeling of semantics and context in the language model, can make it hard for these surface realizers to produce valid syntactic variants in a principled way. In previous work, we built a statistical surface realizer in which word order and word frequency probabilities are folded into the grammar rather than into a second-stage language model (Zhong & Stent 2005). With this surface realizer, we can more effectively model syntactically valid variation.

(Creswell & Kaiser 2004) argue that statistical NLG systems based only on probabilities cannot capture fine distinctions in meaning; discourse context and meaning must be taken into account when selecting a construction for NLG purposes. They demonstrate their point by outlining an algorithm to determine surface form of dative constructions based on manually annotated discourse status (hearer-old and hearer-new) and heaviness of arguments, which would reduce the error rate by 25% compared to surface form as determined by corpus frequency. They argue that labeling data with pragmatic information such as discourse status is crucial for statistical NLG. Bresnan et al. (Bresnan *et al.* 2005) provide evidence for this by showing that some pragmatic features are useful for predicting the dative. We adapt Bresnan's model, but use only automatically extracted features. Unlike (Creswell & Kaiser 2004), we incorporate this model into a statistical NLG system and achieve improved performance on dative constructions.

(Stede 1998) proposed a general approach to handling verb alternation in NLG for English and German. This approach was focused on representing possible alternations rather than on having a surface realizer determine which one to use in a given set of circumstances.

Data

We used two data sets in these experiments; each data set has two parts, one treebanked and one not. The Dialog data set comprises the following two corpora: ROCH, a set of corpora of spoken dialogs collected at the University of Rochester; and the treebanked portion of the Switchboard (SWBD) corpus of spoken dialogs (Marcus, Santorini, & Marcinkiewicz 1995). The Text data set comprises the following two corpora: the APW-Dec-99 portion of the English Gigaword (GW) corpus of raw text (Graff *et al.* 2005); and the Penn Treebank (PT) corpus of parsed text, including the Brown and Wall Street Journal subsets (Marcus, Santorini, & Marcinkiewicz 1995).

Genre	Corpus	Dative examples	
		V-NP-NP	V-NP-PP
Dialog	ROCH	99	274
	SWBD	401	147
Text	PT	818	408
	GW	4067	4117

Table 1: Data used in our experiments

We used automatic methods to prepare each corpus for our experiments as follows. First, we split each treebanked sentence into independent clauses. We parsed non-treebanked sentences into independent clauses using the Collins parser (Collins 1999). Then we automatically annotated each clause with semantic information:

- Verb frames from VerbNet (Kipper, Dang, & Palmer 2000).
- Noun/verb hypernyms from WordNet (Fellbaum 1998). As a backup, we used Comlex (Girshman, Macleod, & Meyers 1994).

We extracted from the treebanked corpora verbs that might be examples of the dative: verbs appearing in sentences of the form V-NP-PP:to with the prepositional phrase's role labeled as -DTV. We augmented this list with the verbs that are in classes with dative frames in VerbNet. Then, from all four of our corpora we automatically extracted clauses containing any of these verbs in either form of the dative alternation.

Table 1 shows the number of clauses in each of our data sets having either the V-NP-NP or the V-NP-PP construction and the number of examples of each form of the dative alternation we were able to extract automatically. We included all examples of either form of the dative, even if in our data a particular verb appeared with only one form.

Because these data are automatically extracted, they include sentences that are not examples of the dative. In particular, in our dialog corpora there are verbs, like 'take', that take the dative alternation and also have a transitive form with a destination; some of these sentences show up in our data if the semantic type of the destination is mislabeled or ambiguous. Also, some incomplete sentences are included in our dialog data set because of sentence splitting errors.

Modeling The Dative Alternation

(Bresnan *et al.* 2005) demonstrated that pragmatic features are not merely redundant with syntactic features when modeling the dative alternation. However, they did not look at which feature types (lexical, syntactic, semantic and pragmatic) are the best approximation to their models, nor at training from noisy data. In this section, we describe how we trained classifiers to address these questions.

Features

We follow the model in (Bresnan *et al.* 2005), with the following adaptations. First, we limit ourselves to features we can extract automatically from raw text, or from automatically annotated (e.g. automatically parsed) text. The model of Bresnan et al. includes certain features, such as givenness and accessibility, that we approximate automatically using Gundel's Givenness Hierarchy (see below). Second, the original model manually grouped verbs into five broad semantic classes, while we use VerbNet classes. Third, we include animacy and person of the theme, which was excluded from the original model due to data sparsity. We also include a feature for concreteness of recipient.

We categorize our features into the following five sets:

1. **Lexical Features**: Pronominality of recipient and theme (T/F).
2. **Syntactic Features**: Number of recipient and theme (sing/plur), Person of recipient and theme (1st/2nd/3rd), Definiteness of recipient and theme (T/F). These are either obtained from the syntactic tree, or using lexical information.
3. **Semantic Features**: Animacy of recipient and theme (T/F/Unknown), Concreteness of recipient and theme (T/F/Unknown), verb frame (from VerbNet). The animacy and concreteness feature values are obtained from WordNet: animacy is T if the hypernym set of the head noun contains "human" or "animal"; concreteness is T if the hypernym set of the head noun contains "physical object"; if the head noun does not have an entry in WordNet, both features will be labeled as unknown. Pronouns are treated separately: animacy and concreteness are both T for first and second person pronouns and third person singular masculine and feminine; for "it" and for third person plural pronouns and deictics, animacy and concreteness are unknown.
4. **Pragmatic Features**: Givenness of theme and recipient(given/new/unknown) and structural parallelism (V-NP-NP/V-NP-PP/NONE). We use an approximation of Gundel's Givenness Hierarchy (Gundel, Hedberg, & Zacharski 1993) to decide the givenness of an NP. We treat In Focus (e.g. pronouns, it), Activated (e.g. this, that, this NN), Familiar (e.g. that NN), Uniquely Identifiable (e.g. the NN) and Referential (e.g. indefinite this NN) as "given", Type Identifiable (a NN) as "new", and all others as "unknown". We include structural parallelism for text and dialog. To approximate the structural parallelism feature, we give the structure of the immediately preceding sentence (text), or of the last utterance in the immediately preceding turn by the other speaker (dialog).
5. **Length Difference Feature** = $length(theme) - length(recipient)$.

Classifier Training

For each experiment discussed below, we trained a set of classifiers to predict alternation. For the purposes of comparing with Bresnan's results, we used logistic regression. We used the implementation provided in Weka (Witten & Frank 2005) with its default parameters. We performed all experiments using ten-fold cross-validation on each data set (dialog and text) separately.

We performed three sets of experiments. We looked at the effect of using different types of feature to predict the dative alternation in dialog and text. We looked at individual features and combinations of features. Finally, we looked at the impact on surface realization of modeling dative alternation.

Experiment 1: Single features

First we looked at how useful individual features were for predicting the dative alternation. We trained binary classifiers to predict the two forms of the dative alternation for both our dialog data set and our text data set. Results are shown in Table 2. Not surprisingly, for both genres, the verb class had the greatest discriminatory power. However, for the dialog data, pronominality of recipient, a lexical feature, also showed good performance, e.g.

Everybody in the world will offer you a credit card.

Everybody in the world will offer a credit card to you.

(The second highest-performing feature for the text data was number of theme, a syntactic feature.)

Experiment 2: Combinations of features

Second, we looked at the predictive power of individual types of feature. We trained binary classifiers for each of our data sets using features from single feature sets. Classification results are shown in Table 3. The most successful classifier for both data sets used semantic information. The lexical feature set also showed relatively high performance, but only for the dialog data. Interestingly, the least successful feature set for the dialog data was the syntactic feature set, while for the text data this feature set gave the second highest performance. The pragmatic feature set was not, on its own, a high performing feature set for either data set. However, this is not particularly surprising, given the difficulty of approximating givenness using automatic means.

Third, we looked at combinations of feature sets. We trained binary classifiers for each combination of feature sets. Results for this experiment are shown in Table 4. For the dialog data, it is clear that the semantic, pragmatic and lexical feature sets are important for obtaining good performance. In fact, higher performance is obtained for a model including just these three feature sets than for a model including all five feature sets. Interestingly, when we added just the givenness features (and not the structural parallelism feature) we obtained no performance improvement over just using lexical and semantic feature sets. For the text data, on the other hand, models that included only the lexical and/or pragmatic feature sets performed poorly, while those that combined te syntactic feature set with the semantic feature set gave good performance. In addition, the best-performing

Data	Features								
	1. Lexical		2. Syntactic						3. Semantic
	pronominality		number		person		definiteness		verb-class
	theme	recipient	theme	recipient	theme	recipient	theme	recipient	
Dialog	70.14	**75.24**	58.31	61.54	54.90	63.41	63.74	52.23	**76.00**
Text	55.54	57.11	64.16	52.71	52.20	51.91	53.21	56.30	**75.89**

Data	Features							
	3. Semantic				4. Pragmatic			5. Length
	animacy		concreteness		givenness		structural	difference
	theme	recipient	theme	recipient	theme	recipient	parallelism	
Dialog	69.52	67.63	71.08	55.06	63.19	72.10	54.25	71.88
Text	52.62	53.88	56.72	53.82	57.84	58.13	65.87	55.88

Table 2: Classification results: single features

Data	Feature Sets				
	1: Lexical	2: Syntactic	3: Semantic	4: Pragmatic	5: Length
Dialog	77.09	69.16	**84.10**	74.26	71.88
Text	57.35	65.71	**77.85**	60.13	55.88

Table 3: Classification results: single feature sets

model for the text data set was the one that included all feature sets.

Experiment 3: Impact on surface realization

We conducted an experiment to explore the impact on surface realization of modeling dative alternation. We compared the performance of a statistical surface realizer when generating dative sentences with and without using our classifiers. For this experiment, we generated sentences using the statistical surface realizer described in (Zhong & Stent 2005). This is a one-stage surface realizer that uses a probabilistic tree-adjoining grammar derived automatically from a training corpus. The input is a semantic logical form, which may contain the lexical items to be used during surface realization. Our surface realizer generates sentences by selecting syntactic structures for each part of the input semantic representation. Figure 1 shows a sample input to our system. For this input, the surface realizer might find two top-level S-trees, one NP-tree for the Agent, two NP-trees for the Recipient, one NP-tree for the Theme and two trees for the main verb. It would instantiate these trees and combine them using adjunction and substitution, giving at least eight possible outputs, ranked according to their likelihood given the training corpus.

For our experiment, we combined all the sentences from all dialog corpora into one data set, and all the sentences from all text corpora into another data set. We kept aside all the examples of the dative that we had extracted from each corpus. We used the rest of the data to train our statistical surface realizer. This means that our statistical surface realizer had no S-rules for the V-NP-PP dative construction, so we added this rule by hand and assigned to it a probability corresponding to the overall relative frequency of this construction in our corpora.

As our baseline, we used our original surface realization

```
(:sentence
(:agent (:hyper person
        :value (:pronoun you :number sing))
:verb (:hyper supply
      :verb-class give-13.1-1
      :value give :number sing
      :tense present :perfect -
      :progressive - :modal can)
:theme (:hyper social-control
        :pronoun false :det false
        :number SING :person 3
        :animacy false :concreteness false
        :status unknown :value punishment
        :adj capital))
:recipient (:hyper person
           :pronoun true :det true
           :number SING :person 3
           :animacy true :concreteness true
           :status old :value (:pronoun he))))
```

Figure 1: Sample input for the sentence "You can give him capital punishment."

algorithm. In our alternative, we used classifiers to determine which S-tree should be used rather than simply trying all of them (an approach similar to that used in Amalgam (Ringger, Corston-Oliver, & Moore 2002) for German). For each data set, we used the highest-performing feature set from experiment 2 to train the classifiers. Using tenfold cross-validation, we repeatedly trained our classifiers on 90% of our dative sentences and tested on the remaining 10%.

We specified content words and prepositions, but no other function words, in the input. We did not construct the logical forms for out input by hand, but used the same procedure we use to create training data for our surface realizer to con-

Data	Feature Sets									
	1+2	1+3	1+4	1+5	2+3	2+4	2+5	3+4	3+5	4+5
Dialog	84.51	87.32	77.32	77.09	87.54	81.79	*70.56*	**88.32**	86.99	72.08
Text	69.54	77.85	60.88	*58.04*	79.02	68.96	68.18	77.85	**79.84**	65.99

Data	Feature Sets									
	1+2 +3	1+2 +4	1+2 +5	1+3 +4	1+3 +5	1+4 +5	2+3 +4	2+3 +5	2+4 +5	3+4 +5
Dialog	87.43	87.24	84.51	**89.88**	87.10	*77.32*	88.88	88.54	86.59	88.65
Text	81.13	71.09	**82.37**	80.31	81.72	*66.85*	81.40	81.40	73.56	80.31

Data	Feature Sets					
	1+2 3+4	1+2 3+5	1+2 4+5	1+3 4+5	2+3 4+5	all
Dialog	89.21	88.88	*87.13*	**90.10**	88.88	89.10
Text	82.14	82.85	*74.56*	81.96	83.13	**83.60**

Table 4: Classification results: combinations of feature sets. Feature set mappings are: 1. lexical, 2. syntactic, 3. semantic, 4. pragmatic and 5. length.Highest performing combinations are boldface; lowest performing are italicized.

Genre	Without Classifiers		With Classifiers	
	SSA	F-measure	SSA	F-measure
Dialog	60.54	84.99	85.10	88.23
Text	67.66	84.6	83.67	86.51

Table 5: Surface realization results

struct the logical forms for our input.

Our generation results are shown in Table 5. We report performance using simple string accuracy (SSA), a widely-used evaluation metric for surface realization, and Melamed's F, which is not as susceptible to sentence length differences as SSA and which does not punish as much for movement (Turian, Shen, & Melamed 2003).

The surface realizer with classifiers outperformed the one without classifiers when generating dative sentences. This result is statistically significant for both data sets (paired t-tests, $\alpha < .001$). Table 6 shows some output sentences from each surface realizer.

Furthermore, after the classifier was loaded into memory, time-to-generate was almost the same as without the classifier (a difference of less than 5 milliseconds per sentence), so this approach to generation would work very well for dialog.

This is an interesting result, particularly in the case of our dialog data set, because dialog interactions are highly context-dependent.

Discussion

Our overall results confirm Creswell and Kaiser's (2004) hypothesis that some generation decisions cannot be made without semantic knowledge. However, we also found interesting differences between text and dialog; in particular, that lexical and pragmatic features are more helpful for dialog than for text. Conversation is more contextually grounded so there are typically more pronouns and other contextually dependent reference, which are included in both the lexical and pragmatic feature sets (e.g. (Bresnan *et al.* 2005) report that in the Wall Street Journal corpus for the recipient argument, nouns outnumber pronouns 5 to 1, but in the treebanked Switchboard recipient pronouns outnumber nouns almost 4 to 1). On the other hand, the syntactic features are more helpful for text. In text, sentence structure is typically more formal and complex than in dialog.

Because we did not annotate our data by hand, we do not have sufficient evidence for the importance of contextual and pragmatic information. One possible place where both types of information would be important is in constructing corrections, e.g.

A: John gave Sally the box
B: No, John gave it to me

It would be interesting to explore the impact of additional pragmatic features, such as topic/task, presence and type of discourse cue in the sentence, and (for the dialog data) type of dialog act.

Conclusions and Future Work

At the beginning of this paper, we set out to do three things: determine if a linguistic model such as that in (Bresnan *et al.* 2005) can be useful when adapted with automatically extracted features; examine which feature types have the greatest discriminatory power for different types of language (text and dialog) when impoverished models for predicting the dative alternation must be used; and examine the impact on surface realization of modeling dative alternation.

Our results show that the dative alternation can be successfully modeled with noisy data – automatically extracted sentences and automatically labeled features. The best-performing models differ slightly between text and dialog; but, for both types of language, these models improve gen-

Data Set	Without Classifiers	With Classifiers
Dialog	*there offers educational experience to people* *you can give capital punishment to him*	*there offers people educational experience* *you can give him capital punishment*
Text	*we can send the offender a message* *the facility provides military personnel and veterans services*	*we can send a message to the offender* *the facility provides services to military personnel and veterans*

Table 6: Example sentences generated by our statistical surface realizer with or without classifiers for predicting the form of the dative

eration performance for dative constructions. Since most corpora seem to contain limited use of the dative alternation, this result alone is not likely to lead to significantly improved surface realization except in certain domains (pertaining, for example, to financial transactions). However, we believe that similar techniques can be productively applied to other patterns of syntactic variation that depend on a complex array of factors.

We are currently exploring the application of machine learning to other types of syntactic variation that may be affected by semantic and pragmatic information, including prepositional phrase attachment and adverbials.

References

Baker, M. 1997. *Elements of Grammar*. Kluwer. chapter Thematic roles and syntactic structure, 73–137.

Baptist, L., and Seneff, S. 2000. Genesis-II: a versatile system for language generation in conversational system applications. In *Proceedings of ICSLP 2000*.

Bresnan, J.; Cueni, A.; Nikitina, T.; and Baayen, H. 2005. Predicting the dative alternation. In *Proceedings of the Royal Netherlands Academy of Science Workshop on Foundations of Interpretation*.

Channarukul, S. 1999. Yag: a template-based natural language generator for real-time systems. Master's thesis, University of Wisconsin at Milwaukee.

Collins, P. 1995. The indirect object construction in English: An informational approach. *Linguistics* 33:35–49.

Collins, M. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. Dissertation, University of Pennsylvania.

Creswell, C., and Kaiser, E. 2004. The importance of discourse context for statistical natural language generation. In *Proceedings of the 5th SIGDial Workshop on Discourse and Dialogue*.

Elhadad, M., and Robin, J. 1997. SURGE: a comprehensive plug-in syntactic realization component for text generation. Technical report, Computer Science Department, Ben-Gurion University, Beer Sheva, Israel.

Fellbaum, C. 1998. *WordNet: An electronic lexical database*. MIT Press.

Girshman, R.; Macleod, C.; and Meyers, A. 1994. COMLEX syntax: Building a computational lexicon. In *Proceedings of COLING 94*.

Graff, D.; Kong, J.; Chen, K.; and Maeda, K. 2005. English gigaword second edition. ISBN: 1-58563-350-X.

Gundel, J. K.; Hedberg, N.; and Zacharski, R. 1993. Cognitive status and the form of referring expressions in discourse. *Language* 69:274–307.

Halliday, M. A. K. 1970. Language structure and language function. In Lyons, J., ed., *New Horizons in Linguistics*. Penguin Books.

Harley, H. 2000. Possession and the double object construction. In Pica, P., and Rooryck, J., eds., *Yearbook of Linguistic Variation II*. Amsterdam: John Benjamins Press.

Kipper, K.; Dang, H.; and Palmer, M. 2000. Class-based construction of a verb lexicon. In *Proceedings of AAAI 2000*.

Langkilde, I. 2002. *A foundation for general purpose natural language generation: Sentence realization using probabilistic models of language*. Ph.D. Dissertation, Information Science Institute, University of Southern California.

Marcus, M.; Santorini, B.; and Marcinkiewicz, M. 1995. Treebank-2. ISBN: 1-58563-054-3.

Pesetsky, D. 1995. *Zero syntax: Experiences and cascades*. Cambridge, MA: MIT Press.

Rambow, O., and Bangalore, S. 2000. Using TAGs, a tree model and a language model for generation. In *Proceedings of the Fifth International Workshop on Tree-Adjoining Grammars (TAG+)*.

Ratnaparkhi, A. 2000. Trainable methods for surface natural language generation. In *Proceedings of NAACL 2000*.

Ringger, E.; Corston-Oliver, S.; and Moore, R. 2002. An overview of Amalgam: A machine-learned generation module. In *Proceedings of INLG 2002*.

Stede, M. 1998. A generative perspective on verb alternations. *Computational Linguistics* 24(3).

Thompson, S. 1990. Information flow and dative shift in English discourse. In Edmondson, J. A.; Crawford, F.; and Muhlhausler, P., eds., *Development and Diversity, Language Variation Across Space and Time*. Dallas, Texas: Summer Institute of Linguistics.

Turian, J.; Shen, L.; and Melamed, I. D. 2003. Evaluation of machine translation and its evaluation. In *Proceedings of MT Summit IX*.

Witten, I., and Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques*. San Francisco, CA: Morgan Kaufmann, 2 edition.

Zhong, H., and Stent, A. 2005. Building surface realizers automatically from corpora. In *Proceedings of the Workshop on Using Corpora in Natural Language Generation*.