Classification-based Generation Using TAG

Tomasz Marciniak and Michael Strube

EML Research gGmbH
Villa Bosch Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
http://www.eml-research.de/nlp

Abstract. In this paper we present an application of machine learning to generating natural language route directions. We use the TAG formalism to represent the structure of the generated texts and split the generation process into a number of individual tasks which can be modeled as classification problems. To solve each of these tasks we apply corpus-trained classifiers relying on semantic and contextual features, determined for each task in a feature selection procedure.

1 Introduction

Traditionally, different NLG tasks such as *text structuring*, *lexicalization* or *syntactic realization* have been considered to belong to different problem categories, requiring their own processing methods and representations. In this paper we present a classification-based approach to language generation which affords a uniform treatment of different NLG stages. We decompose the generation process into a sequence of tasks which realize *minimal* elements of the surface grammatical form of an expression, given the meaning to be coded and the realization context. Each task is handled by a separate corpus-trained classifier using semantic and contextual features chosen in a feature selection procedure. To represent the grammatical structure of the generated texts we use the Tree Adjoining Grammar (TAG) formalism which provides an elegant way to account for the syntactic structure of individual sentences as well as the structure of the discourse. We apply our method to generating route directions, focusing on several elements of the grammatical form both at the clause and discourse levels.

The paper is structured as follows: in Section 2 we characterize the semantic and grammatical structure of route directions. In Section 3 we present the classification-based model of generation and in Section 4 we describe our data. The experiments and evaluation are discussed in Section 5 and related work is summarized in Section 6.

2 Route Directions Overview

Route directions are instructional texts providing a detailed specification of the actions that the instructions' recipient should perform in order to reach his/her goal (see Example 1 below). Descriptions of actions (1b, e) are typically accompanied by specifications of *states* (1a) and *events* (1c, d) that may result from the actions or initiate them. As the cover term for actions, states and events we use a single concept of *situation*.

Example 1. (a) Standing in front of the hotel (b) follow Meridian street south for about 100 meters, (c) passing the First Union Bank entrance on your right, (d) until you see the river side in front of you. (e) Then make a left onto North Hills Street.

In our work we focus on the whole process of mapping between the semantic content of route directions and their grammatical form. Hence in this section we discuss meaning elements of the generated texts and present an account of their grammatical structure, which both provide a basis for the generation method described in Section 3.

Table 1. Binary attributes used to specify the aspectual type of a situation

	Binary Representation		
Vendlerian Classes	*Stative*	*Durative*	*Culminated*
States	yes	yes/no	no
Activities	no	yes/no	no
Achievements	no	no	yes
Accomplishments	no	yes	yes

2.1 Semantic Analysis

We analyze the semantic content of instructional texts as comprising three major elements. At the level of individual discourse units it includes the *semantic frame* of the portrayed situation and its *aspectual type*. Furthermore, we associate the discourse-level meaning of such texts with the *temporal structure* of the discourse, based on temporal relations holding between individual situations. The semantic frame provides a schematic representation of a situation, based on its ontological class (e.g. *self-motion*, *visual perception*) which can be further associated with a set of specific semantic roles (e.g. *self mover*, *path* or *goal*) (cf. [1]). The aspectual type of a situation denotes its qualitative temporal structure. To characterize it, we follow the analysis by [2] and associate each situation with three binary attributes (see Table 1). The representation thus obtained allows to discriminate between *Vendlerian* classes of situations [3]. Clause (1b) from the above example tagged with the semantic information is presented below:

$$\left[\ \textit{follow}\ \left[\textit{Meridian Street}\right]_{\text{PATH}} \left[\textit{south}\right]_{\text{DIRECTION}} \left[\ \textit{for about 100 meters}\ \right]_{\text{DISTANCE}}\ \right]_{\text{SELF\_MOTION, –STATIVE, +DURATIVE, –CULMINATED}}$$

The temporal structure of route directions can be modeled as a tree (see Figure 1), with nodes corresponding to discourse units and edges signalling temporal relations. We interpret these relations as holding between the actual situations referred to by each pair of connected nodes. Relation labels that we use include: *initial*, *ongoing* and *subsequent*, which denote the particular time interval of the situation referred to by the parent node during which the situation specified by the child node occurs.

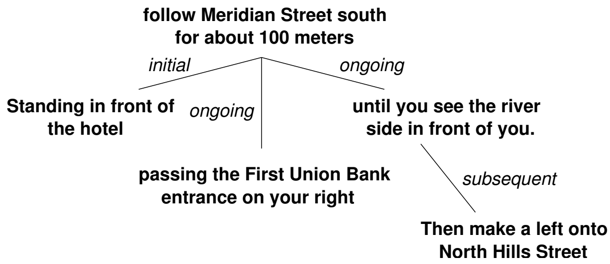

Fig. 1. Temporal Structure

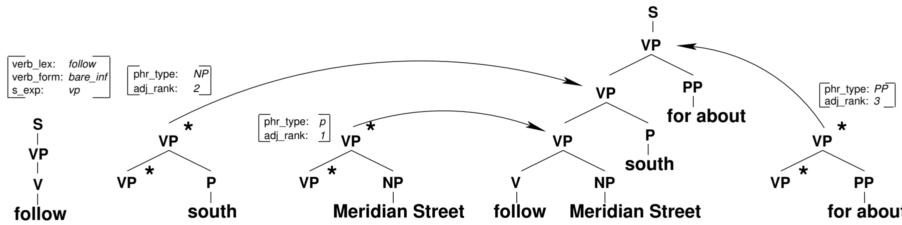

Fig. 2. Elementary trees are represented as feature vectors. Tree selection occurs in a series of classifications which specify individual feature values. Phrasal structures are then adjoined to the tree anchored by the verb in the order determined by one of the features.

2.2 Grammatical Form

To represent the grammatical structure of route directions we use the TAG formalism [4]. A TAG can be defined as a tree rewriting system composed of a set of *elementary* trees which are combined by means of *adjunction* operations to form a derived structure. A selection of elementary trees and the derivation process for clause (1b) are presented in Figure 2.

Note that we do not specify explicitly the subcategorization frame of the verb. All the syntactic patterns necessary to correctly realize a clause are learned directly from the data, so that an explicit model is not necessary. We also abandon the traditional way of augmenting a TAG with semantic information by embedding individual nodes with feature structures (cf. [5], [6]). Instead we acknowledge the fact that virtually *all* elements of the semantic content of an expression may influence realization of *any* element of its grammatical form. We believe that the decision which semantic elements should be considered for each realization step, should be based on empirical grounds.

We base our account of the discourse structure of route directions on DLTAG, a discourse extension to the lexicalized TAG, proposed by [7]. In DLTAG, discourse units function as arguments of *discourse connectives* which anchor elementary trees, labeled as *discourse clauses* (D_c). The derivation process starts with a single initial tree associated with the discourse unit which occupies the root position in the temporal discourse model (cf. Figure 1). All other discourse units are associated with auxiliary trees, which are successively combined to form a derived structure (see Figure 3).

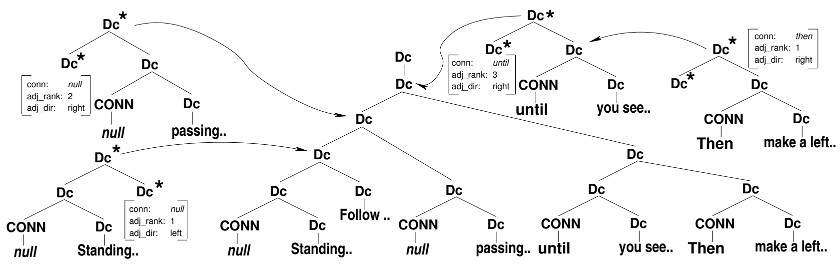

Fig. 3. Selection of elementary trees is based on feature values by means of which individual trees are represented. If two trees are to be adjoined at the same root node, their order is specified by their adjunction rank.

3 Classification-Based Generation

Linguistic realization of an expression is a process involving several steps, in which *minimal* elements of the grammatical structure are assigned specific forms given the meaning to be coded and the context. We define these *minimal* elements as distinctive components of the structure which, depending on the particular form they take, may change the meaning of an expression or render it ill-formed. In the TAG-based representation that we adopted, minimal elements correspond to those elements of a tree which allow to differentiate between two elementary structures. They may include structural elements, such as the adjunction direction, or lexical ones, e.g. the tree anchor. Each such element is modeled as a distinctive feature in the vector representation of elementary trees (see Table 2).

In the approach to generation presented here we formulate realizations of individual features as single classification problems. The entire generation process is then modeled as a cascade of classifiers. As its input, each classifier takes a selection of semantic and contextual features. The first group of features is directly based on the analysis of route directions presented in Section 2. Contextual features specify how the same task has been solved in the preceding discourse, and what forms have been assigned to other elements within the same discourse unit. We narrowed this type of context to the two immediately preceding discourse units. Corresponding feature names have been prefixed with *n_* and *nn_* respectively. We also used the numerical features *null_prec* and *related_count*, which specify respectively how many discourse units with a *null* connective precede, and how many discourse units are related to the current one.

The architecture of our system consists of a cascade of eight classifiers which incrementally build up a vector representation of the expression's form. As their input

Table 2. Vector representation of elementary trees

Tree Anchor	Feature	Description	Possible Values
discourse connective	*conn*	Connective Lexical Form	*and, until, null, . . .*
	adj_rank_dsc	Adjunction Rank	numeric
	adj_dir	Adjunction Direction	right, left
main verb	*s_exp*	S Node Constituents	np_vp, vp
	verb_lex	Verb Lexical Form	*walk, follow, turn, . . .*
	verb_form	Verb Form	gerund, bare_inf, finite_pres, . . .
verb argument	*phr_type*	Phrase Type	np, pp, p
	adj_rank_phr	Adjunction Rank	numeric

the classifiers take a semantic specification of an expression arranged in a feature vector. In addition, those classifiers which occur later in the pipeline may take the output of previous modules as part of their own input. To determine which semantic features should be used at each realization stage, we applied a feature selection procedure based on the *wrapper approach* [8]. We used the same method to decide in which order individual classifiers should be placed, as this determines the availability of the contextual information at each realization stage. The ordering of tasks and the selected subsets of features are presented in Table 3.

Table 3. Feature subsets used by individual classifiers. Tasks *adj_rank_dsc* and *adj_rank_phr* are further split into a series of binary classifications considering each possible pair of constituents which are to be ordered. Hence two subsets of these features are considered for each binary task.

Task	Semantic Features	Contextual Features
adj_rank_dsc	*relation, frame, action, stative, durative*	*related_count*
adj_dir	*relation ,n_relation, frame, action, durative, culminated*	*adj_rank_dsc, related_count*
connective	*relation, n_relation, action, stative, durative, culminated, n_durative*	*adj_rank_dsc, adj_dir, n_connective, nn_connective, related_count, null_prec*
s_exp	*frame, relation, action, stative, durative, culminated*	*adj_rank_dsc, adj_dir, connective*
verb_form	*relation, action, stative, durative, culminated*	*adj_rank_dsc, adj_dir, connective, s_exp, n_verb_form, nn_verb_form, related_count*
verb_lex	*relation, frame, action, stative, durative, culminated, source_rel, path_rel, goal_rel, dir_rel*	*connective, verb_form, n_verb_lex, related_count*
phr_type	*sem_role, arg_rel*	*verb_lex, s_exp, verb_form,*
adj_rank_phr	*sem_role, arg_rel*	*phr_type, verb_lex, s_exp, verb_form, connective*

4 Data

Our corpus consists of 70 manually annotated texts, obtained from printed tourist guide books or found on the Internet. Individual texts have a similar structure and their length ranges from 8 to 17 discourse units realized as clauses, yielding a total of 916 clauses in the entire corpus. The annotations comprise markables, i.e. marked text spans falling in four different groups. *Discourse-unit* markables relate to individual situation descriptions and were tagged with attributes specifying the semantic frame of a situation and its aspectual structure. To indicate temporal relations holding between discourse units,

Table 4. Results

Task	Classes	Majority Baseline	Rule-based baseline Accuracy	F-M.	KStar Accuracy	F-M.	Count
1. *adj_rank_dsc*	*numeric*	-	65.38%	-	87.96%	-	916
1a *precedence*	yes	50%	71.03%	0.76	93.35%	0.93	466
	no			0.64		0.93	466
2. *adj_dir*	right	90.61%	98.25%	0.99	97.82%	0.99	830
	left			0.89		0.89	86
3. *connective*	null	64.52%	63.86%	0.76	79.91%	0.87	591
	and			0.21		0.61	160
	until			0.61		0.86	57
	after			0.54		0.76	31
	as			0.56		0.62	32
	...			...		...	...
4. *s_exp*	vp	78.38%	92.85%	0.94	94.76%	0.97	718
	np_vp			0.82		0.88	198
5. *verb_form*	bare_inf	59.39%	76.31%	0.87	91.05%	0.97	544
	gerund			0.56		0.86	128
	fin_pres			0.78		0.87	169
	will_inf			0.58		0.63	68
	to_inf			0		0.6	7
6. *verb_lex*	walk	13.10%	32.17%	0.26	71.83%	0.65	120
	turn			0.42		0.89	104
	pass			0.45		0.88	66
	follow			0.34		0.64	53
	continue			0.17		0.54	53
	...			...		...	...
7. *phr_type*	pp	40.10%	83.28%	0.84	92.16%	0.94	573
	np			0.83		0.92	667
	p			0.80		0.87	189
8. *adj_rank_phr*	*numeric*	-	66.15%	-	83.96%	-	1429
8a *precedence*	yes	50%	72.26%	0.73	91.06%	0.91	682
	no			0.72		0.91	682

markables at this level were combined with directed links labeled with specific relation names. If a pair of related discourse units was connected by means of a conjunction or an adverbial, they were tagged as *Discourse-connective* markables. If no explicit discourse connective was found, a *null* connective was assumed. Main verbs within each clause were tagged as *Situation-predicate* markables. Arguments of verbs were tagged as *Situation-argument* markables and were associated with specific semantic roles. All grammatical types of information, such as *verb form* or *constituent type* of verb arguments, were determined automatically during a post-processing stage.

To obtain training and test data from the annotated corpus we used the following procedure. First, we represented each text in the form of a tree (as in Figure 1) . Each node was then described by a vector of feature values, specifying its semantic content and the grammatical structure of the respective discourse unit. We used such constructed discourse representations to generate instances for each classification task, using each time the relevant subset of the semantic features and obtaining the contextual features from the information available at the neighboring nodes.

5 Experiments and Results

To evaluate our system we performed a series of experiments, using an instance-based classifier KStar [9], implemented in Weka [10]. We chose this learning scheme as it performed better on our data than the rule induction algorithm Ripper [11] and the decision-tree learner C4.5 [12].

The results reported in Table 4 were obtained using 10-fold cross-validation for each task. For comparison we also provide the scores of two baseline schemes. They include a *majority* baseline which always selects a class label with the highest frequency in the training data, and a manually-constructed rule-based system which considers up to four semantic and contextual features for each task (see Figure 4). A brief summary of each task and the results are presented below:

Adjunction Rank / Discourse (1, 1a). This numeric task is split into a series of binary *precedence* classifications which consider each pair of elementary trees and decide on their relative order. Results of these sub-tasks are used to sort the relevant trees and assign them a unique rank. The rule-based system reaches 71.03% accuracy at the binary *precedence* task and 65.38% at the main numeric task. The accuracy of KStar at the binary classification lies at 93.35% which yields 87.96% accuracy at the overall adjunction rank task.

Adjunction Direction (2). The majority baseline for this binary task is 90.61%. It is the only task where the rule-based system outperforms the learning algorithm. It assigns *"right"* as a default to all instances with *local rank* higher than *1* or *relation* other than *initial*. Note that the instance-based learner considers more features, which were chosen in the feature selection procedure.

Discourse Connective (3). For this multi-class task the majority baseline lies relatively hight at 64.52% (associated with the *null* connective). The accuracy of the rule-based system is lower here, 63.86%, but it reaches higher F-Measure for other connective classes. KStar proves much better here than the baseline schemes, reaching the accuracy of 79.91%. Table 5 (left) presents a fragment of the confusion matrix with the five most frequent classes. Most misclassifcations occur within narrow groups of classes which in some context may signal similar relations, e.g. *null* vs. *and*, or *as* vs. *after*.

Table 5. Fragments of confusion matrices for *connective*, *verb_form* and *verb_lex* classes

null	*and*	*until*	*after*	*as*	*classified ← as*
523	94	11	4	4	*null*
60	94	2	0	1	*and*
4	0	53	0	0	*until*
2	0	0	26	3	*after*
7	0	0	7	18	*as*

bare_inf	*ger*	*fin_pres*	*will_inf*	*classified ← as*
536	6	1	1	*bare_inf*
21	103	4	0	*gerund*
3	5	146	12	*fin_pres*
6	1	21	38	*will_inf*

walk	*turn*	*pass*	*continue*	*classified ← as*
86	2	2	3	*walk*
0	98	0	2	*turn*
1	0	61	0	*pass*
14	0	0	28	*continue*

S Expansion (4). This binary task specifies the main constituent structure of a clause: *NP +VP* vs. *VP* (i.e. if it has no subject). In our domain the majority of clauses is subjectless (78.38%). This task is solved comparatively well by both the rule-based scheme and our system, which reach 92.85% and 94.76% accuracy, respectively.

Verb Form (5). The majority baseline for this task lies at 59.39% (*bare_inf*) and the accuracy of the rule-based system is 76.31%. KStar reaches a much higher score, 91.05%, and improves on the F-Measure for individual classes. Two classes which got lower F-Measure are *to_inf* which has a very low frequency and *will_inf*. The relevant confusion matrix is shown in Table 5 (middle).

Verb Lex (6). This task is concerned with choosing the lexical form of the main verb. It is a multi-class problem, with majority baseline relatively low at 13.10% (*walk*). The overall accuracy of the rule based system reaches 32.17%. The instance-based classifier performs much better here, scoring 71.83% accuracy. A fragment of the confusion matrix for this task is presented in Table 5 (right).

Phrase Type (7). At this task, the phrase type of the verb argument is determined. The majority baseline lies at 40.10% (*PP*). The rule based system performs reasonably well on this task, scoring 83.28%. Our system reaches 92.16% accuracy here.

Adjunction Rank / Phrase (8, 8a). Similarly to (1), this task is split into a series of binary classifications. The rule based system reaches 72.26% at the binary and 66.15% at the main task. Results of our system are much better here: 91.06% and 83.96%.

```
if (relation == subsequent)
    if (null_prec == 2)
        conn = and;
    else
        conn = null;
else if (relation == ongoing)
       if (action == yes)
           conn = null;
       else
           conn = until;
else if (relation == subsequent)
       ...
```

```
if (action == yes)
    verb_form = bare_inf;
else if (relation == subsequent)
    verb_form = fin_pres;
else if (relation == ongoing)
    if (connective == until)
        verb_form = fin_pres;
    else
        verb_form = gerund;
else if (relation == initial)
    ...
```

Fig. 4. Baseline hand-crafted rules for *connective* (left) and *verb_form* (right) tasks

6 Related Work

Empirical methods were introduced to NLG in the context of syntactic realization [13]. Most current works in this area follow the *ranking* approach which involves overgeneration and then selection of the best candidate, e.g. [14], [15]. Different corpus-based techniques were applied at the discourse planning stage to fact ordering, e.g. [16], [17]. While each of the aforementioned works focused on individual tasks, [18] presented an attempt to merge two trainable modules for different processing stages, which we also do in our work. In contrast to the above works we apply a single trainable method to different NLG tasks, trying to span several stages of the generation process. Our classification model also performs candidate ranking but it considers elements of the grammatical structure and not whole sentences, hence overgeneration is avoided.

The idea that different processing stages in an NLP system can be represented as classification tasks and solved using supervised machine learning methods was stated in [19]. So far this approach was used in NLG for solving individual tasks only [20, 16].

7 Conclusions

In this paper we presented our work on generating route directions, which is a part of a larger project concerned with producing natural language output from a *tourist information system*. We modeled several NLG stages in terms of classification problems which can be solved using machine learning methods. The advantage of this approach is the uniform treatment of different NLG tasks which facilitates adding new modules to the system and re-training it for novel domains.

We found that for almost all tasks machine learning techniques proved much better than hand-crafted rules. This can be explained by the fact that the simple heuristics that we applied considered no more than four information sources (i.e. *features*) at a time, whereas the machine-learning classifiers took advantage of a much more fine-grained instance representation. Arguably, hand-crafting classification rules which take a larger feature space into consideration would require an extensive problem expertise so that the procedure could not be easily replicated for new tasks or domains.

One issue which still requires considerable thought is the evaluation method. Information-retrieval scores such as F-Measure are too strict for many NLG tasks, especially those involving lexical choice. Such tasks can be characterized as *multi-label* problems, in which identical patterns in the data may be associated with different class labels. This is exactly the case in natural language, where more than one lexical or grammatical construction may be appropriate in a given context. Also, in the current work we limited the evaluation to individual tasks only, skipping the overall assessment of the generated discourse. We believe that discourse-level evaluation should be based on human judgements of text coherence and quality. We also want to follow [21] and look for correlations between human qualitative judgements and quantitative measures.

Finally, we plan to extend the scope of our work to cover the remaining NLG tasks (esp. content selection and referring expression generation), and focus on other text sorts in the domain of *tourist information*.

Acknowledgements: The work presented here has been funded by the Klaus Tschira Foundation, Heidelberg, Germany.

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics, *Montréal, Québec, Canada, 10–14 August 1998*. (1998) 86–90
2. Moens, M., Steedman, M.: Temporal ontology and temporal reference. Computational Linguistics **14** (1988) 15–28
3. Vendler, Z.: Verbs and times. In: Linguistics in Philosophy, *Cornell University Press, Ithaca, NY*. (1967) 97–121
4. Joshi, A.K., Schabes, Y.: Tree-adjoining grammars and lexicalized grammars. In: Maurice Nivat and Andreas Podelski, editors, *Definability and Recognizability of Sets of Trees*. Elsevier. (1991)

5. Stone, M., Doran, C.: Sentence planning as description using tree adjoining grammar. In: Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, *Madrid, Spain, 7 – 12 July 1997*. (1997) 198–205
6. Bleam, T., Palmer, M., Shanker, V.: Motion verbs and semantic features in TAG. In: Proceedings of the 4th International Workshop on Tree-adjoining Grammars and Related Frameworks (TAG+), *Philadelphia, PA., 1 – 3 August, 1998*. (1998)
7. Webber, B.L., Joshi, A.: Anchoring a lexicalized tree-adjoining grammar for discourse. In: Proceedings of the COLING/ACL '98 Workshop on Discourse Relations and Discourse Markers, *Montréal, Québec, Canada, 15 August 1998*. (1998) 86–92
8. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence Journal **97** (1997) 273–324
9. Cleary, J.G., Trigg, L.E.: K*: An instance-based learner using an entropic distance measure. In: Proceedings of the 12th International Conference on Machine Learning, *Tahoe City, Ca., 9–12 July, 1995*. (1995) 108–114
10. Witten, I.H., Frank, E.: Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco, CA (2000)
11. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the 12th International Conference on Machine Learning, *Tahoe City, Ca., 9–12 July, 1995*. (1995) 115–123
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco, CA (1993)
13. Langkilde, I., Knight, K.: Generation that exploits corpus-based statistical knowledge. In: Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics, *Montréal, Québec, Canada, 10–14 August 1998*. (1998) 704–710
14. Bangalore, S., Rambow, O.: Exploiting a probabilistic hierarchical model for generation. In: Proceedings of the 18th International Conference on Computational Linguistics, *Saarbrücken, Germany, 31 July – 4 August 2000*. (2000) 42–48
15. Varges, S., Mellish, C.: Instance-based natural language generation. In: Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics, *Pittsburgh, PA, 2 – 7 June, 2001*. (2001) 1–8
16. Duboue, P.K., McKeown, K.R.: Empirically estimating order constraints for content planning in generation. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, *Toulouse, France, 9–11 July 2001*. (2001) 172–179
17. Dimitromanolaki, A., Androutsopoulos, I.: Learning to order facts for discourse planning in natural language generation. In: Proceedings of the 9th European Workshop on Natural Language Generation, *Budapest, Hungary, 13 – 14 April 2003*. (2003) 23–30
18. Chen, J., Bangalore, S., Rambow, O., Walker, M.: Towards automatic generation of natural language generation systems. In: Proceedings of the 19th International Conference on Computational Linguistics, *Taipei, Taiwan, 24 August – 1 September, 2002*. (2002)
19. Daelemans, W., van den Bosch, A.: Rapid development of NLP modules with memory-based learning. In: Proceedings of ELSNET in Wonderland, pp. 105-113. Utrecht: ELSNET, 1998. (1998) 105–113
20. Hardt, D., Rambow, O.: Generation of VP ellipsis: A corpus-based approach. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, *Toulouse, France, 9–11 July 2001*. (2001) 282–289
21. Bangalore, S., Rambow, O., Whittaker, S.: Evaluation metrics for generation. In: Proceedings of the 1st International Natural Language Generation Conference, *Mitzpe Ramon, Israel, 12 – 16 June 2000*. (2000) 1–8