Collective Content Selection for Concept-To-Text Generation

Regina Barzilay
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
regina@csail.mit.edu

Mirella Lapata
School of Informatics
University of Edinburgh
mlap@inf.ed.ac.uk

Abstract

A content selection component determines which information should be conveyed in the output of a natural language generation system. We present an efficient method for automatically learning content selection rules from a corpus and its related database. Our modeling framework treats content selection as a collective classification problem, thus allowing us to capture contextual dependencies between input items. Experiments in a sports domain demonstrate that this approach achieves a substantial improvement over context-agnostic methods.

1 Introduction

Content selection is a fundamental task in concept-to-text generation (Reiter and Dale, 2000). A practical generation system typically operates over a large database with multiple entries that could potentially be included in a text. A content selection component determines what subset of this information to include in the generated document.

For example, consider the task of automatically generating game summaries, given a database containing statistics on Americal football. Table 1 shows an excerpt from such a database, and its corresponding game summary written by a journalist. A single football game is typically documented in hundreds of database entries — all actions, player positions, and scores are recorded, along with a wide range of comparative and aggregate statistics. Only a small fraction of this information is featured in a game summary. The content selection component aims to identify this subset.[1]

In existing generation systems the content selection component is manually crafted. Specifying content selection rules is, however, notoriously difficult, prohibitively so in large domains. It involves the analysis of a large number of texts from a domain-relevant corpus, familiarity with the associated database, and consultation with domain experts. Moreover, the task must be repeated for each domain anew.

This paper proposes a data-driven method for learning the content-selection component for a concept-to-text generation system. We assume that the learning algorithm is provided with a parallel corpus of documents and a corresponding database, in which database entries that should appear in documents are marked.

One possible approach is to formulate content selection as a standard binary classification task: predict whether an item is to be included on the basis of its attributes alone. In fact, this method is commonly used for content selection in text summarization (e.g., Kupiec et al., 1995). However, by treating each instance in isolation, we cannot guarantee that the selected database entries are related in a meaningful way, which is essential for the generation of a coherent text.

Rather than selecting each item separately, we propose a method for *collective content selection*, where all candidates are considered simultaneously for selection. Collective selection thereby allows us to explicitly optimize *coherence* in the generated

[1] The organization of the selected information and its surface realization is typically handled by other components of the generation system, which are outside the scope of this paper.

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 331–338, Vancouver, October 2005.

Passing					
PLAYER	CP/AT	YDS	AVG	TD	INT
Brunell	17/38	192	6.0	0	0
Garcia	14/21	195	9.3	1	0
...	...	...	...	...	...

Rushing					
PLAYER	REC	YDS	AVG	LG	TD
Suggs	22	82	3.7	25	1
...	...	...	...	...	...

Fumbles				
PLAYER	FUM	LOST	REC	YDS
Coles	1	1	0	0
Portis	1	1	0	0
Davis	0	0	1	0
Little	0	0	1	0
...	...	...	...	...

Suggs rushed for 82 yards and scored a touchdown in the fourth quarter, leading the Browns to a 17-13 win over the Washington Redskins on Sunday. **Jeff Garcia went 14-of-21 for 195 yards and a TD** for the Browns, who didn't secure the win until **Coles fumbled** with 2:08 left. The Redskins (1-3) can pin their third straight loss on going just 1-for-11 on third downs, mental mistakes and **a costly fumble by Clinton Portis**. **Brunell finished 17-of-38 for 192 yards**, but was unable to get into any rhythm because Cleveland's defense shut down Portis. The Browns faked a field goal, but holder Derrick Frost was stopped short of a first down. **Brunell then completed a 13-yard pass to Coles, who fumbled** as he was being taken down and Browns safety Earl Little recovered.

Table 1: Sample target game description and example of database entries; boldface indicates correspondences between the text and the database (CP/AT: completed out of attempted, YDS: yards, AVG: average, TD: touchdown, INT: interception, REC: received, LG: longest gain, FUM: fumble).

text: semantically related entries are often selected together. In essence, the algorithm seeks a subset of candidates that is consistent with the individual preferences of each candidate, and at the same time maximally satisfies contextual constraints. A graph-based formulation of this optimization problem allows us to find an exact, globally optimal solution, using a min-cut algorithm.

Collective content selection is particularly beneficial to generation systems that operate over relational databases. Rich structural information available in a database can be readily utilized to determine semantic relatedness between different database entries. For instance, we can easily find all actions (e.g., touchdowns and fumbles) associated with a specific player in a game, which could be relevant for generating a summary centered around an individual. We show how to utilize database relations for discovering meaningful contextual links between database entries.

We evaluate our collective content selection model in a sports domain. The proposed content selection component operates over a large database containing descriptive statistics about American football games. Our model yields a 10% increase in F-score, when compared to a standard classification approach, thus demonstrating the benefits of collective content selection on this complex domain. Furthermore, our results empirically confirm the contribution of discourse constraints for content selection.

In the following section, we provide an overview of existing work on content selection. Then, we define the learning task and introduce our approach for collective content selection. Next, we present our experimental framework and data. We conclude the paper by presenting and discussing our results.

2 Related Work

The generation literature provides multiple examples of content selection components developed for various domains (Kukich, 1983; McKeown, 1985; Sripada et al., 2001; Reiter and Dale, 2000). A common theme across different approaches is the emphasis on coherence: related information is selected "to produce a text that hangs together" (McKeown, 1985). Similarly, our method is also guided by coherence constraints. In our case these constraints are derived automatically, while in symbolic generation systems coherence is enforced by analyzing a large number of texts from a domain-relevant corpus and

careful hand-crafting of content selection rules.

Duboue and McKeown (2003) were the first to propose a method for learning content selection rules automatically, thus going beyond mere corpus analysis. They treat content selection as a classification task. Given a collection of texts associated with a domain-specific database, their model learns whether a database entry should be selected for presentation or not. Their modeling approach uses an expressive feature space while considering database entries in isolation.

Similarly to Duboue and McKeown (2003), we view content selection as a classification task and learn selection rules from a database and its corresponding corpus. In contrast to them, we consider all database entries simultaneously, seeking a globally optimal selection. Thus, we avoid the need for extensive feature engineering by incorporating discourse constraints into the learning framework. In addition, we assess whether data-driven methods for content selection scale up to large databases with thousands of interrelated entries, by evaluating our model in a sports domain. Previous work (Duboue and McKeown, 2003) has tackled the content selection problem for biographical summaries, a simpler domain with fewer entities and interactions among them.

3 The Task

We assume that the content selection component takes as input a set of database entries.[2] Each entry has a type and a set of attributes associated with its type. For instance, the database shown in Table 1 contains entries of three types — `Passing`, `Rushing` and `Fumbles`. Two entries are of type `Passing`, and each of them has six attributes — `PLAYER`, `CP/AT`, `YDS`, `AVG`, `TD`, `INT`. In addition, each entry has a label that specifies whether it should be included in a generated text or not.

During the training process, the learning algorithm is provided with n sets of database entries, each associated with a label whose value is known. In practice, we only require a parallel corpus of game summaries and database entries — label values are derived automatically via alignment (see Section 4 for more details).

[2] A terminological note: a database entry is analogous to a row in a relational table; throughout this paper we use the terms entity and database entry interchangeably.

The goal of the content selection component is to select entries from a database, i.e., to determine whether their label values are 0 or 1. Under this formulation, content selection is restricted to information available in the database; there is no attempt to induce new facts through inference.

In the next section, we describe our learning framework, and explain how it is applied to the content selection task.

3.1 The Collective Classification Approach

Generation of a coherent text crucially depends on our ability to select entities that are related in a meaningful way (McKeown, 1985). A content selection component that considers every entity in isolation does not have any means to enforce this important discourse constraint. We therefore formulate content selection as a *collective classification* task, where all entities that belong to the same database (i.e., the same football game) are considered simultaneously. This framework thus enables us to enforce contextual constraints by selecting related entities.

When considered in isolation, some database entries are more likely to be selected than others. In the American football domain, for example, entries of type `Rushing` are often extracted if they yield a touchdown.[3] Other `Rushing` entries (e.g., which do not deliver scoring points) are typically omitted. In general, the attributes of an entry can provide useful cues for predicting whether it should be selected. Therefore, we can perform content selection by applying a standard classifier on each entry. In Section 3.2, we explain in more detail how such a classifier can be trained.

We can also decide about entity selection by analyzing how entities relate to each other in the database. For instance, in a game where both quarterbacks[4] score, it is fairly unorthodox to mention the passing statistics for only one of them. Label assignments in which either both quarterbacks are selected, or both of them are omitted should be there-

[3] A touchdown is the primary method of scoring in American football; a touchdown is worth six points and is accomplished by gaining legal possession of the ball in the opponent's end zone.

[4] A quarterback in American football is the leader of a team's offense. In most offenses his primary duty is passing the ball. Quarterbacks are typically evaluated on their passing statistics, including total yardage, completion ratio, touchdowns, and the ability to avoid interceptions.

fore preferred. This relation between quarterback passing statistics exemplifies one type of link that can hold between entities. Other link types may encode contextual constraints, for instance capturing temporal and locational information. (In Section 3.3, we describe a method for discovering link types which encapsulate meaningful contextual dependencies.) By taking into account links between related entities, a content selection component can enforce dependencies in the labeling of related entities.

Our goal is to select a subset of database entities that maximally satisfies linking constraints and is as consistent as possible with the individual preferences of each entity. Thus, content selection can be naturally stated as an optimization problem — we wish to find a label assignment that *minimizes* the cost of violating the above constraints.

Let C_+ and C_- be a set of selected and omitted entities, respectively; $ind_+(x)$ and $ind_-(x)$ are scores that capture the individual preference of x to be either selected or omitted, and $link_L(x,y)$ reflects the degree of dependence between the labels of x and y based on a link of type L. Thus, the optimal label assignment for database entries $x_1,\ldots,x_n$ will minimize:

$$\sum_{x\in C_+} ind_-(x) + \sum_{x\in C_-} ind_+(x) + \sum_L \sum_{\substack{x_i\in C_+ \\ x_j\in C_-}} link_L(x_i,x_j)$$

The first two elements in this expression capture the penalty for assigning entities to classes against their individual preferences. For instance, the penalty for selecting an entry $x \in C_+$ will equal $ind_-(x)$, i.e., x's individual preference of being ommitted. The third term captures a linking penalty for all pairs of entities (x_i,x_j) that are connected by a link of type L, and are assigned to different classes.

This formulation is similar to the energy minimization framework, which is commonly used in image analysis (Besag, 1986; Boykov et al., 1999) and has been recently applied in natural language processing (Pang and Lee, 2004). The principal advantages of this formulation lie in its computational properties. Despite seeming intractable — the number of possible subsets to consider for selection is exponential in the number of database entities — the inference problem has an exact solution. Provided that the scores $ind_+(x)$, $ind_-(x)$, and $link_L(x,y)$ are positive, we can find a globally optimal label assignment in polynomial time by computing a minimal cut partition in an appropriately constructed graph (Greig et al., 1989).

In the following we first discuss how individual preference scores are estimated. Next, we describe how to induce links and estimate their scores.

3.2 Computing Individual Preference Scores

The individual preference scores are estimated by considering the values of entity attributes, recorded in the database. The type and number of the attributes are determined by the entity type. Therefore, we separately estimate individual preference scores for each entity type. For example, individual scores for entities of type `Passing` are computed based on six attributes : `PLAYER`, `CP/AT`, `YDS`, `AVG`, `TD`, `INT` (see Table 1).

Considerable latitude is available when selecting a classifier for delivering the individual preference scores. In our experiments we used the publicly available BoosTexter system (Schapire and Singer, 2000). BoosTexter implements a boosting algorithm that combines many simple, moderately accurate categorization rules into a single, highly accurate rule. For each example, it outputs a prediction along with a weight whose magnitude indicates the classifier's confidence in the prediction. We thus set the individual preference scores to the weights obtained from BoosTexter. The weights range from -1 to 1; we obtained non-negative numbers, simply by adding 1.

It is important to note that BoosTexter is a fairly effective classifier. When applied to text categorization (Schapire and Singer, 2000), it outperformed a number of alternative classification methods, including Naive Bayes, decision trees, and k-nearest neighbor.

3.3 Link Selection and Scoring

The success of collective classification depends on finding links between entities with similar label preferences. In our application — concept-to-text generation, it is natural to define entity links in terms of their database relatedness. Since the underlying database contains rich structural information, we can explore a wide range of relations between database entities.

The problem here is finding a set of links that

capture important contextual dependencies among many possible combinations. Instead of manually specifying this set, we propose a corpus-driven method for discovering links automatically. Automatic link induction can greatly reduce human effort. Another advantage of the method is that it can potentially identify relations that might escape a human expert and yet, when explicitly modeled, aid in content selection.

We induce important links by adopting a generate-and-prune approach. We first automatically create a large pool of candidate links. Next, we select only links with aconsistent label distributions.

Construction of Candidate Links An important design decision is the type of links that we allow our algorithm to consider. Since our ultimate goal is the generation of a coherent text, we wish to focus on links that capture semantic connectivity between database entities. An obvious manifestation of semantic relatedness is attribute sharing. Therefore, we consider links across entities with one or more shared attributes. An additional constraint is implied by computational considerations: our optimization framework, based on minimal cuts in graphs, supports only pairwise links, so we restrict our attention to binary relations.

We generate a range of candidate link types using the following template: For every pair of entity types E_i and E_j, and for every attribute k that is associated with both of them, create a link of type $L_{i,j,k}$. A pair of entities $\langle a,b\rangle$ is linked by $L_{i,j,k}$, if a is of type E_i, b is of type E_j and they have the same value for the attribute k. For example, a link that associates statistics on `Passing` and `Rushing` performed by the same player is an instantiation of the above with $E_i =$ Rushing, $E_j =$ Passing, and $k =$ Player.

In a similar fashion, we construct link types that connect together entities with two or three attributes in common. Multiple pairs of entries can be connected by the same link type.

If the database consists of n entity types, and the number of attribute types is bounded by m, then the number of link types constructed by this process does not exceed $O(n^2(m+\binom{m}{2}+\binom{m}{3})) \approx O(n^2m^3)$. In practice, this bound is much lower, since only a few attributes are shared among entity types. Links can be efficiently computed using SQL's `SELECT` operator.

Link Filtering Only a small fraction of the automatically generated link types will capture meaningful contextual dependencies. To filter out spurious links, we turn to the labels of the entities participating in each link. Only link types in which entities have a similar distribution of label values are selected from the pool of candidates.

We measure similarity in label distribution using the χ^2 test. This test has been successfully applied to similar tasks, such as feature selection in text classification (Rogati and Yang, 2002), and can be easily extended to our application. Given a binary link, our null hypothesis H_0 is that the labels of entities related by L are independent. For each link, we compute the χ^2 score over a 2-by-2 table that stores joint label values of entity pairs, computed across all database entries present in the training set. For links with $\chi^2 > \tau$, the null hypothesis is rejected, and the link is considered a valid discourse constraint. The value of τ is set to 3.84, which corresponds to a 5% level of statistical significance.

Link Weights The score of a link type L is defined as follows:

$$link_L(x,y) = \begin{cases} \lambda_L & if(x,y) \text{ are linked by L} \\ 0 & \text{otherwise} \end{cases}$$

We estimate link weights λ_L using simulated annealing. The goal is to find weight values that minimize an objective function, defined as the error rate on the development set[5] (see Section 4 for details). The individual scores and the link structure of the entities in the development set are predicted automatically using the models trained on the training set. Starting from a random assignment of weight values, we compute the objective function and generate new weight values using Parks' (1990) method. The procedure stops when no sufficient progress is observed in subsequent iterations.

4 Evaluation Framework

We apply the collective classification method just presented to the task of automatically learning content selection rules from a database containing football-related information. In this section, we first present the sport domain we are working with, and

[5]Our objective function cannot be optimized analytically. We therefore resort to heuristic search methods such as simulated annealing.

Entity Type	Attr	Inst	%Aligned	Entity Type	Attr	Inst	%Aligned
Defense	8	14,077	0.00	*Passing*	5	1,185	59.90
Drive	10	11,111	0.00	*Team comparison*	4	14,539	0.00
Play-by-Play	8	83,704	3.03	*Punt-returns*	8	940	5.74
Fumbles	8	2,937	17.78	*Punting*	9	950	0.87
Game	6	469	0.00	*Receiving*	8	6,337	11.19
Interceptions	6	894	45.05	*Rushing*	8	3,631	9.17
Kicking	8	943	26.93	*Scoring-sum*	9	3,639	53.34
Kickoff-returns	8	1,560	5.24	*Team*	3	4	0.00
Officials	8	464	0.00				

Table 2: Entity types and their attributes in the NFL database; percentage of database entries that are aligned to summary sentences.

describe how we collected a corpus for evaluating collective content selection. Next, we explain how we automatically obtained annotated data for training and testing our model.

Data As mentioned previously our goal is to generate descriptions of football games. The sports domain has enjoyed popularity among natural language generation practitioners (Robin, 1994; Tanaka-Ishii et al., 1998). The appeal is partly due to the nature of the domain — it exhibits several fixed patterns in content organization and is therefore amenable to current generation approaches. At the same time, it is complex enough to present challenges at almost all stages of the generation process.

We compiled a corpus of descriptions of football games from the web. More specifically, we obtained game summaries from the official site of the American National Football League[6] (NFL). We collected summaries for the 2003 and 2004 seasons. These are typically written by Associated Press journalists. The corpus consists of 468 texts in total (436,580 words). The average summary length is 46.8 sentences.

The site not only contains a summary for each game, but also a wealth of statistics describing the performance of individual players and their teams. It includes a scoring summary and a play-by-play summary giving details of the most important events in the game together with temporal (i.e., time remaining) and positional (i.e., location in the field) information. In sum, for each game the site offers a rich repository of tabulated information which we translated into a relational database. An excerpt of the database is shown in Table 1. Table 2 displays the entity types contained in our NFL database and lists the number of attributes (Attr) and instantiations (Inst) per type. The database contains 73,400 entries in total.

Alignment Recall that our collective classification method is supervised. The training instances are database entries and the class labels indicate whether an instance should be selected for presentation or not. We could obtain this information via manual annotation performed by domain experts. Instead, we opted for a less costly, automatic solution that yields large quantities of training and testing data. To infer which database entries correspond to sentences in the verbalized game summaries, we used a simple anchor-based alignment technique. In our domain, numbers and proper names appear with high frequency, and they constitute reliable anchors for alignment. Similar to previous work (Duboue and McKeown, 2003; Sripada et al., 2001), we employ a simple matching procedure that considers anchor overlap between entity attributes and sentence tokens.

Overall, the alignment procedure produced 7,513 pairs. 7.1% of the database entries were verbalized in our corpus and 31.7% of the corpus sentences had a database entry. Table 2 presents the proportion of database entries which are verbalized in our corpus, broken down by entity type (see %Aligned).

To evaluate the accuracy of this procedure, we compared our output with a gold-standard alignment produced by a domain expert. After analyzing the data from five games, the expert produced 52 alignment pairs; 47 of these pairs were identified

[6] See `http://www.nfl.com/scores`.

	Majority Baseline			Standard Classifier			Collective Classifier		
	Prec	Rec	F-score	Prec	Rec	F-score	Prec	Rec	F-score
Mean	29.40	68.19	40.09	44.88	62.23	49.75	**52.71**	**76.50**	**60.15**
Min	3.57	28.57	6.45	12.50	8.33	13.33	12.50	27.27	19.05
Max	57.14	100.00	65.12	76.92	100.00	75.00	100.00	100.00	100.00
Std Dev	10.93	15.75	12.25	15.36	18.33	13.98	21.29	18.93	19.66

Table 3: Results on content selection (precision, recall and F-score are averages over individual game summaries); comparison between the majority baseline, standard and collective classification.

by the automatic alignment. In addition, three pairs produced by the program did not match the gold-standard alignment. Thus, the automatic method achieved 94.0% precision and 90.4% recall.

Data Annotation For training and testing purposes, we only considered entity types for which alignments were observed in our corpus (e.g., `Fumbles`, `Interceptions`; see Table 2). Types without alignments can be trivially regarded as inappropriate for selection in the generated text. We considered database entries for which we found verbalizations in the corpus as positive instances (i.e., they should be selected); accordingly, non-verbalized entries were considered negative instances (i.e., they should not be selected). The overall dataset contained 105,792 instances (corresponding to 468 game summaries). Of these, 15% (68 summaries) were reserved for testing. We held out 1,930 instances (10 summaries) from the training data for development purposes.

5 Results

Our results are summarized in Table 3. We compare the performance of the collective classifier against a standard classifier. This can be done in our framework, simply by setting the link scores to zero. We also report the performance of a majority baseline. The latter was obtained by defaulting to the majority class for each entity type in the training data. As can be seen from Table 2, only for two relations — `Passing` and `Scoring-sum` — the majority class predicts that the corresponding database instances should be selected for presentation.

Our results confirm that a content selection component can be automatically engineered for the football domain. The collective classifier achieves an F-score of 60.15%. This result compares favorably with Duboue and McKeown (2003) whose best model has an F-score of 51.00% on a simpler domain. Our method has high recall (we want to avoid missing out information that should be presented in the output) but tends to overgenerate as demonstrated by the relatively moderate precision in Table 3. Erroneous content selection decisions could be remedied by other components later in the generation process. Alternatively, the obtained content selection rules could be further refined or post-processed by a domain expert. Finally, better classification performance should be possible with more expressive feature sets. As we can see from the weak performance of the standard classifier, attribute values of database entries may not be sufficiently strong predictors. Considering additional features tailored to the NFL domain could further enhance performance. However, feature selection is not one of the main objectives of this work.

Our results empirically validate the importance of discourse constraints for content selection (Table 4 illustrates examples of constraints that the model discovered). We observe that adding contextual information leads to a 10.4% F-score increase over the standard classifier. We used a paired t test to examine whether the differences are statistically significant. The collective model significantly outperforms the standard model on both precision ($t = 4.824$, $p < 0.01$) and recall ($t = 8.445$, $p < 0.01$). It is also significantly better than the majority baseline, both in terms of recall ($t = 3.181$, $p < 0.01$) and precision ($t = 8.604$, $p < 0.01$). The standard classifier performs significantly better than the majority baseline on precision ($t = 7.043$, $p < 0.01$) but worse on recall ($t = -2.274$, $p < 0.05$).

6 Conclusions and Future Work

In this paper we have presented a novel, data-driven method for automating content selection. Central

$\{\langle a,b\rangle \mid a \in Sum \wedge b \in Sum \wedge a.Quarter = b.Quarter\}$
$\{\langle a,b\rangle \mid a \in Sum \wedge b \in Play \wedge Sum.Player_1 = Play.Player_1 \wedge Sum.Action = Play.Action\}$
$\{\langle a,b\rangle \mid a \in Fumbles \wedge b \in Interceptions \wedge Fumbles.Player = Interceptions.Player\}$

Table 4: Examples of automatically derived links.

to our approach is the use of a collective classification model that captures contextual dependencies between input items. We show that incorporation of discourse constraints yields substantial improvement over context-agnostic methods. Our approach is linguistically grounded, computationally efficient, and viable in practical applications.

In the future, we plan to explore how to integrate more refined discourse models in the content selection process. Currently, we consider a limited set of contextual dependencies based on attribute similarity. Ideally, we would like to express more complex relations between items. For instance, we may want to represent disjunctive constraints, such as "at least one of the defense players should be mentioned in the summary." Such dependencies can be efficiently handled in a collective classification framework by using approximate probabilistic inference (Taskar et al., 2002). Another promising approach is the combination of our automatically acquired cross-entity links with domain knowledge.

Needless to say, content selection is one of several components within a working generation system. An interesting question is how to integrate our component into a generation pipeline, using feedback from other components to guide collective content selection.

Acknowledgments

The authors acknowledge the support of the National Science Foundation (Barzilay; CAREER grant IIS-0448168 and grant IIS-0415865) and EPSRC (Lapata; grant GR/T04540/01). We are grateful to Eli Barzilay for his help with data collection, and Luke Zettelmoyer who explained the many rules of American football to us. Thanks to Michael Collins, Amit Dubey, Noemie Elhadad, Dina Katabi, Frank Keller, Igor Malioutov, Smaranda Muresan, Martin Rinard, Kevin Simler and the anonymous reviewers for helpful comments and suggestions. Any opinions, findings, and conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of the National Science Foundation or EPSRC.

References

J. Besag. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48:259–302.

Y. Boykov, O. Veksler, R. Zabih. 1999. Fast approximate energy minimization via graph cuts. In *ICCV*, 377–384.

P. A. Duboue, K. R. McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In *Proceedings of the EMNLP*, 121–128.

D. Greig, B. Porteous, A. Seheult. 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 51(2):271–279.

K. Kukich. 1983. Design of a knowledge-based report generator. In *Proceedings of the ACL*, 145–150.

J. Kupiec, J. O. Pedersen, F. Chen. 1995. A trainable document summarizer. In *Proceedings of the SIGIR*, 68–73.

K. R. McKeown. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

B. Pang, L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, 271–278, Barcelona, Spain.

G. Parks. 1990. An intelligent stochastic optimization routine for nuclear fuel cycle design. *Nuclear Technology*, 89:233–246.

E. Reiter, R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

J. Robin. 1994. *Revision-Based Generation of Natural Language Summaries Providing Historical Background*. Ph.D. thesis, Columbia University.

M. Rogati, Y. Yang. 2002. High-performing feature selection for text classification. In *Proceedings of the CIKM*, 659–661.

R. E. Schapire, Y. Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

S. G. Sripada, E. Reiter, J. Hunter, J. Yu. 2001. A two-stage model for content determination. In *Proceedings of the ACL-ENLG*, 3–10.

K. Tanaka-Ishii, K. Hasida, I. Noda. 1998. Reactive content selection in the generation of real-time soccer commentary. In *Proceedings of the ACL/COLING*, 1282–1288.

B. Taskar, P. Abbeel, D. Koller. 2002. Discriminative probabilistic models for relational data. In *Proceedings of the UAI*, 485–495.