Discourse Generation Using Utility-Trained Coherence Models

Radu Soricut
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
radu@isi.edu

Daniel Marcu
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
marcu@isi.edu

Abstract

We describe a generic framework for integrating various stochastic models of discourse coherence in a manner that takes advantage of their individual strengths. An integral part of this framework are algorithms for searching and training these stochastic coherence models. We evaluate the performance of our models and algorithms and show empirically that utility-trained log-linear coherence models outperform each of the individual coherence models considered.

1 Introduction

Various theories of discourse coherence (Mann and Thompson, 1988; Grosz et al., 1995) have been applied successfully in discourse analysis (Marcu, 2000; Forbes et al., 2001) and discourse generation (Scott and de Souza, 1990; Kibble and Power, 2004). Most of these efforts, however, have limited applicability. Those that use manually written rules model only the most visible discourse constraints (e.g., the discourse connective "although" marks a CONCESSION relation), while being oblivious to fine-grained lexical indicators. And the methods that utilize manually annotated corpora (Carlson et al., 2003; Karamanis et al., 2004) and supervised learning algorithms have high costs associated with the annotation procedure, and cannot be easily adapted to different domains and genres.

In contrast, more recent research has focused on stochastic approaches that model discourse coherence at the local lexical (Lapata, 2003) and global levels (Barzilay and Lee, 2004), while preserving regularities recognized by classic discourse theories (Barzilay and Lapata, 2005). These stochastic coherence models use simple, non-hierarchical representations of discourse, and can be trained with minimal human intervention, using large collections of existing human-authored documents. These models are attractive due to their increased scalability and portability. As each of these stochastic models captures different aspects of coherence, an important question is whether we can combine them in a model capable of exploiting all coherence indicators.

A frequently used testbed for coherence models is the discourse ordering problem, which occurs often in text generation, complex question answering, and multi-document summarization: given N discourse units, what is the most coherent ordering of them (Marcu, 1996; Lapata, 2003; Barzilay and Lee, 2004; Barzilay and Lapata, 2005)? Because the problem is NP-complete (Althaus et al., 2005), it is critical how coherence model evaluation is intertwined with search: if the search for the best ordering is greedy and has many errors, one is not able to properly evaluate whether a model is better than another. If the search is exhaustive, the ordering procedure may take too long to be useful.

In this paper, we propose an A* search algorithm for the discourse ordering problem that comes with strong theoretical guarantees. For a wide range of practical problems (discourse ordering of up to 15 units), the algorithm finds an optimal solution in reasonable time (on the order of seconds). A beam search version of the algorithm enables one to find good, approximate solutions for very large reordering tasks. These algorithms enable us not only to compare head-to-head, for the first time, a set of coherence models, but also to combine these models so as to benefit from their complementary strengths. The model com-

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 803–810,
Sydney, July 2006. ©2006 Association for Computational Linguistics

bination is accomplished using statistically well-founded utility training procedures which automatically optimize the contributions of the individual models on a development corpus. We empirically show that utility-based models of discourse coherence outperform each of the individual coherence models considered.

In the following section, we describe previously-proposed and new coherence models. Then, we present our search algorithms and the input representation they use. Finally, we show evaluation results and discuss their implications.

2 Stochastic Models of Discourse Coherence

2.1 Local Models of Discourse Coherence

Stochastic local models of coherence work under the assumption that well-formed discourse can be characterized in terms of specific distributions of local recurring patterns. These distributions can be defined at the lexical level or entity-based levels.

Word-Coocurrence Coherence Models. We propose a new coherence model, inspired by (Knight, 2003), that models the intuition that the usage of certain words in a discourse unit (sentence) tends to trigger the usage of other words in subsequent discourse units. (A similar intuition holds for the Machine Translation models generically known as the IBM models (Brown et al., 1993), which assume that certain words in a source language sentence tend to trigger the usage of certain words in a target language translation of that sentence.)

We train models able to recognize local recurring patterns of word usage across sentences in an unsupervised manner, by running an Expectation-Maximization (EM) procedure over pairs of consecutive sentences extracted from a large collection of training documents[1]. We expect EM to detect and assign higher probabilities to recurring word patterns compared to casually occurring word patterns.

A local coherence model based on IBM Model 1 assigns the following probability to a text T consisting of n sentences $s_1, s_2, \ldots, s_n$:

$$P_{\mathrm{IBM}_1^{\mathrm{D}}}(T) = \Pi_{i=1}^{n-1}\Pi_{j=1}^{|s_{i+1}|}\frac{\epsilon}{|s_i|+1}\Sigma_{k=0}^{|s_i|}t(s_{i+1}^j|s_i^k)$$

We call the above equation the direct IBM Model 1, as this model considers the words in sentence s_{i+1} (the s_{i+1}^j events) as being generated by the words in sentence s_i (the s_i^k events, which include the special s_i^0 event called the NULL word), with probability $t(s_{i+1}^j|s_i^k)$. We also define a local coherence inverse IBM Model 1:

$$P_{\mathrm{IBM}_1^{\mathrm{I}}}(T) = \Pi_{i=1}^{n-1}\Pi_{k=1}^{|s_i|}\frac{\epsilon}{|s_{i+1}|+1}\Sigma_{j=0}^{|s_{i+1}|}t(s_i^k|s_{i+1}^j)$$

This model considers the words in sentence s_i (the s_i^k events) as being generated by the words in sentence s_{i+1} (the s_{i+1}^j events, which include the special s_{i+1}^0 event called the NULL word), with probability $t(s_i^k|s_{i+1}^j)$.

Entity-based Coherence Models. Barzilay and Lapata (2005) recently proposed an entity-based coherence model that aims to learn abstract coherence properties, similar to those stipulated by Centering Theory (Grosz et al., 1995). Their model learns distribution patterns for transitions between discourse entities that are abstracted into their syntactic roles – subject (**S**), object (**O**), other (**X**), missing (**-**). The feature values are computed using an entity-grid representation for the discourse that records the syntactic role of each entity as it appears in each sentence. Also, salient entities are differentiated from casually occurring entities, based on the widely used assumption that occurrence frequency correlates with discourse prominence (Morris and Hirst, 1991; Grosz et al., 1995). We exclude the coreference information from this model, as the discourse ordering problem cannot accommodate current coreference solutions, which assume a pre-specified order (Ng, 2005).

In the jargon of (Barzilay and Lapata, 2005), the model we implemented is called Syntax+Salience. The probability assigned to a text $T = s_1 \ldots s_n$ by this Entity-Based model (henceforth called EB) can be locally computed (i.e., at sentence transition level) using M feature functions, as follows:

$$P_{\mathrm{EB}}(T) = \Pi_{i=1}^{n-1}\Sigma_{k=1}^{M}w_k e_k(s_{i+1}|s_i)$$

Here, $e_k(s_{i+1}|s_i)$ are feature values, and w_k are weights trained to discriminate between coherent, human-authored documents and examples assumed to have lost some degree of coherence (scrambled versions of the original documents).

2.2 Global Models of Discourse Coherence

Barzilay and Lee (2004) propose a document content model that uses a Hidden Markov Model

[1] We use for training the publicly-available GIZA++ toolkit, http://www.fjoch.com/GIZA++.html

(HMM) to capture more global aspects of coherence. Each state in their HMM corresponds to a distinct “topic”. Topics are determined by an unsupervised algorithm via complete-link clustering, and are written as h_i, with $h_i \in H$.

The probability assigned to a text $T = s_1 \ldots s_n$ by this Content Model (henceforth called CM) can be written as follows:

$$P_{\mathrm{CM}}(T) = \max_{h_1 \ldots h_n} \Pi_{i=1}^{n} P_{\mathrm{TT}}(h_i | h_{i-1}) \times P_{\mathrm{TM}}(s_i | h_i)$$

The first term, P_{TT}, models the probability of changing from topic h_{i-1} to topic h_i. The second term, P_{TM}, models the probability of generating sentences from topic h_i.

2.3 Combining Local and Global Models of Discourse Coherence

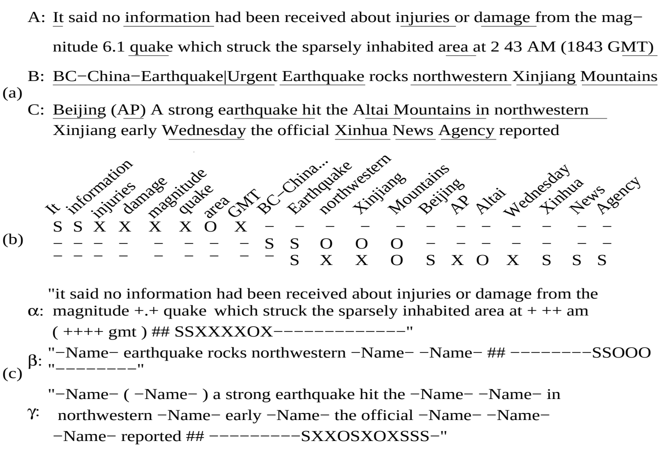
We can model the probability $P(T)$ of a text T using a log-linear model that combines the discourse coherence models presented above. In this framework, we have a set of M feature functions $f_m(T)$, $1 \leq m \leq M$. For each feature function, there exists a model parameter λ_m, $1 \leq m \leq M$. The probability $P(T)$ can be written under the log-linear model as follows:

$$P(T) = \frac{\exp[\Sigma_{m=1}^{M} \lambda_m f_m(T)]}{\Sigma_{T'} \exp[\Sigma_{m=1}^{M} \lambda_m f_m(T')]}$$

Under this model, finding the most probable text T is equivalent with solving Equation 1, and therefore we do not need to be concerned about computing expensive normalization factors.

$$\arg\max_{T} P(T) = \arg\min_{T} -\Sigma_{m=1}^{M} \lambda_m \log f_m(T) \quad (1)$$

In this framework, we distinguish between the *modeling* problem, which amounts to finding appropriate feature functions for the discourse coherence task, and the *training* problem, which amounts to finding appropriate values for λ_m, $1 \leq m \leq M$. We address the modeling problem by using as feature functions the discourse coherence models presented in the previous sections. In Section 3, we address the training problem by performing a discriminative training procedure of the λ_m parameters, using as utility functions a metric that measures how different a training instance is from a given reference.



(a)

A: It said no information had been received about injuries or damage from the mag− nitude 6.1 quake which struck the sparsely inhabited area at 2 43 AM (1843 GMT)

B: BC−China−Earthquake|Urgent Earthquake rocks northwestern Xinjiang Mountains

C: Beijing (AP) A strong earthquake hit the Altai Mountains in northwestern Xinjiang early Wednesday the official Xinhua News Agency reported

(b)

It	information	injuries	damage	magnitude	quake	area	GMT	BC−China...	Earthquake	northwestern	Xinjiang	Mountains	Beijing	AP	Altai	Wednesday	Xinhua	News	Agency
S	S	X	X	X	X	O	X	−	−	−	−	−	−	−	−	−	−	−	−
−	−	−	−	−	−	−	−	S	S	O	O	O	−	−	−	−	−	−	−
−	−	−	−	−	−	−	−	−	S	X	X	O	S	X	O	X	S	S	S

(c)

α: "it said no information had been received about injuries or damage from the magnitude +.+ quake which struck the sparsely inhabited area at + ++ am (++++ gmt) ## SSXXXXOX−−−−−−−−−−−−"

β: "−Name− earthquake rocks northwestern −Name− −Name− ## −−−−−−−−SSOOO "−−−−−−−−"

γ: "−Name− (−Name−) a strong earthquake hit the −Name− −Name− in northwestern −Name− early −Name− the official −Name− −Name− −Name− reported ## −−−−−−−−−SXXOSXOXSSS−"

Figure 1: Example consisting of discourse units A, B, and C (a). In (b), their entities are detected (underlined) and assigned syntactic roles: **S** (subject), **O** (object), **X** (other), **-** (missing). In (c), terms α, β, and γ encode these discourse units for model scoring purposes.

3 Search Algorithms for Coherent Discourses and Utility-Based Training

The algorithms we propose use as input representation the IDL-expressions formalism (Nederhof and Satta, 2004; Soricut and Marcu, 2005). We use here the IDL formalism (which stands for Interleave, Disjunction, Lock, after the names of its operators) to define finite sets of possible discourses over given discourse units. Without losing generality, we will consider sentences as discourse units in our examples and experiments.

3.1 Input Representation

Consider the discourse units A-C presented in Figure 1(a). Each of these units undergoes various processing stages in order to provide the information needed by our coherence models. The entity-based model (EB) (Section 2), for instance, makes use of a syntactic parser to determine the syntactic role played by each detected entity (Figure 1(b)). For example, the string SSXXXXOX------------ (first row of the grid in Figure 1(b), corresponding to discourse unit A) encodes that It and information have subject (S) role, injuries, etc. have other (X) roles, area has object (O) role, and the rest of the entities do not appear (-) in this unit.

In order to be able to solve Equation 1, the input representation needs to provide the necessary information to compute all f_m terms, that is, all individual model scores. Textual units A, B,

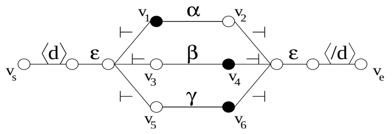


Figure 2: The IDL-graph corresponding to the IDL-expression $\langle d \rangle \cdot \|(\alpha, \beta, \gamma) \cdot \langle /d \rangle$.

and C in our example are therefore represented as terms α, β, and γ, respectively[2] (Figure 1(c)). These terms act like building blocks for IDL-expressions, as in the following example:

$$E = \langle d \rangle \cdot \|(\alpha, \beta, \gamma) \cdot \langle /d \rangle$$

E uses the $\|$ (Interleave) operator to create a bag-of-units representation. That is, E stands for the set of all possible order permutations of α, β, and γ, with the additional information that any of these orders are to appear between the beginning $\langle d \rangle$ and end of document $\langle /d \rangle$. An equivalent representation, called IDL-graphs, captures the same information using vertices and edges, which stand in a direct correspondence with the operators and atomic symbols of IDL-expressions. For instance, each $\vdash$ and $\dashv$–labeled edge k-pair, and their source and target vertices, respectively, correspond to a k-argument $\|$ operator. In Figure 2, we show the IDL-graph corresponding to IDL-expression E.

3.2 Search Algorithms

Algorithms that operate on IDL-graphs have been recently proposed by Soricut and Marcu (2005). We extend these algorithms to take as input IDL-graphs over non-atomic symbols (such that the coherence models can operate inside terms like α, β, and γ from Figure 1), and also to work under models with hidden variables such as CM (Section 2.2).

These algorithm, called IDL-CH-A* (A* search for IDL-expressions under Coherence models) and IDL-CH-HB^b (Histogram-Based beam search for IDL-expressions under Coherence models, with histogram beam b), assume an alphabet Σ of non-atomic (visible) variables (over which the input IDL-expressions are defined), and an alphabet H of hidden variables. They *unfold* an input IDL-graph on-the-fly, as follows: starting from the initial vertex v_s, the input graph is traversed in an IDL-specific manner, by creating states which keep track of k positions in any subgraph corresponding to a k-argument $\|$ operator, as well as the last edge traversed and the last hidden variable considered. For instance, state S $=$ $(\mathrm{v}_1\mathrm{v}_4\mathrm{v}_6, \gamma, h_i)$ (see the blackened vertices in Figure 2) records that expressions β and γ have already been considered (while α is still in the future of state S), and γ was the last one considered, evaluated under the hidden variable h_i. The information recorded in each state allows for the computation of a current coherence cost under any of the models described in Section 2. In what follows, we assume this model to be the model from Equation 1, since each of the individual models can be obtained by setting the other λs to 0.

We also define an *admissible heuristic* function (Russell and Norvig, 1995), which is used to compute an admissible future cost a for state S, using the following equation:

$$a(S) = -\sum_{e \in F} \sum_{m=1}^{M} \lambda_m \min_{\substack{h_i \in H \\ \langle ce, h_j \rangle \in C \times H}} \log f_m(\langle e, h_i \rangle | \langle ce, h_j \rangle)$$

F is the set of future (visible) *events* for state S, which can be computed directly from an input IDL-graph, as the set of all Σ–edge-labels between the vertices of state S and final vertex v_e. For example, for state S $=$ $(\mathrm{v}_1\mathrm{v}_4\mathrm{v}_6, \gamma, h_i)$, we have $F = \{\alpha, \langle /d \rangle\}$. C is the set of future (visible) *conditions* for state S, which can be obtained from F (any non-final future event may become a future conditioning event), by eliminating $\langle /d \rangle$ and adding the current conditioning event of S. For the considered example state S, we have $C = \{\alpha, \gamma\}$. The value $a(S)$ is admissible because, for each future event $\langle e, h_i \rangle$, with $e \in F$ and $h_i \in H$, its cost is computed using the most inexpensive conditioning event $\langle ce, h_j \rangle \in C \times H$.

The IDL-CH-A* algorithm uses a priority queue Q (sorted according to total cost, computed as current $+$ admissible) to control the unfolding of an input IDL-graph, by processing, at each unfolding step, the most inexpensive state (extracted from the top of Q). The admissibility of the future costs and the monotonicity property enforced by the priority queue guarantees that IDL-CH-A* finds an *optimal* solution to Equation 1 (Russell and Norvig, 1995).

The IDL-CH-HB^b algorithm uses a histogram beam b to control the unfolding of an input IDL-graph, by processing, at each unfolding step, the

[2] Following Barzilay and Lee (2004), proper names, dates, and numbers are replaced with generic tokens.

top b most inexpensive states (according to total cost). This algorithm can be tuned (via b) to achieve good trade-off between speed and accuracy. We refer the reader to (Soricut, 2006) for additional details regarding the optimality and the theoretical run-time behavior of these algorithms.

3.3 Utility-based Training

In addition to the modeling problem, we must also address the training problem, which amounts to finding appropriate values for the λ_m parameters from Equation 1.

The solution we employ here is the discriminative training procedure of Och (2003). This procedure learns an optimal setting of the λ_m parameters using as optimality criterion the utility of the proposed solution. There are two necessary ingredients to implement Och's (2003) training procedure. First, it needs a search algorithm that is able to produce ranked k-best lists of the most promising candidates in a reasonably fast manner (Huang and Chiang, 2005). We accommodate k-best computation within the IDL-CH-HB^{100} algorithm, which decodes bag-of-units IDL-expressions at an average speed of 75.4 sec./exp. on a 3.0 GHz CPU Linux machine, for an average input of 11.5 units per expression.

Second, it needs a criterion which can automatically assess the quality of the proposed candidates. To this end, we employ two different metrics, such that we can measure the impact of using different utility functions on performance.

TAU (Kendall's τ). One of the most frequently used metrics for the automatic evaluation of document coherence is Kendall's τ (Lapata, 2003; Barzilay and Lee, 2004). TAU measures the minimum number of adjacent transpositions needed to transform a proposed order into a reference order. The range of the TAU metric is between -1 (the worst) to 1 (the best).

BLEU. One of the most successful metrics for judging machine-generated text is BLEU (Papineni et al., 2002). It counts the number of unigram, bigram, trigram, and four-gram matches between hypothesis and reference, and combines them using geometric mean. For the discourse ordering problem, we represent hypotheses and references by index sequences (e.g., "4 2 3 1" is a hypothesis order over four discourse units, in which the first and last units have been swapped with respect to the reference order). The range of BLEU scores is between 0 (the worst) and 1 (the best).

We run different discriminative training sessions using TAU and BLEU, and train two different sets of the λ_m parameters for Equation 1. The log-linear models thus obtained are called Log-linear$_{\mathsf{maxTAU}}$ and Log-linear$_{\mathsf{maxBLEU}}$, respectively.

4 Experiments

We evaluate empirically two different aspects of our work. First, we measure the performance of our search algorithms across different models. Second, we compare the performance of each individual coherence model, and also the performance of the discriminatively trained log-linear models. We also compare the overall performance (model & decoding strategy) obtained in our framework with previously reported results.

4.1 Evaluation setting

The task on which we conduct our evaluation is information ordering (Lapata, 2003; Barzilay and Lee, 2004; Barzilay and Lapata, 2005). In this task, a pre-selected set of information-bearing document units (in our case, sentences) needs to be arranged in a sequence which maximizes some specific information quality (in our case, document coherence). We use the information-ordering task as a means to measure the performance of our algorithms and models in a well-controlled setting. As described in Section 3, our framework can be used in applications such as multi-document summarization. In fact, Barzilay et al. (2002) formulate the multi-document summarization problem as an information ordering problem, and show that naive ordering algorithms such as majority ordering (select most frequent orders across input documents) and chronological ordering (order facts according to publication date) do not always yield coherent summaries.

Data. For training and testing, we use documents from two different genres: newspaper articles and accident reports written by government officials (Barzilay and Lapata, 2005). The first collection (henceforth called `EARTHQUAKES`) consists of Associated Press articles from the North American News Corpus on the topic of natural disasters. The second collection (henceforth called `ACCIDENTS`) consists of aviation accident reports from the National Transportation Safety

Search Algorithm	IBM_1^{D}			IBM_1^{I}			CM			EB		
	ESE	TAU	BLEU	ESE	TAU	BLEU	ESE	TAU	BLEU	ESE	TAU	BLEU
	EARTHQUAKES											
IDL-CH-A*	0%	.39	.12	0%	.33	.13	0%	.39	.12	0%	.19	.05
IDL-CH-HB^{100}	0%	.38	.12	0%	.32	.13	0%	.39	.12	0%	.19	.06
IDL-CH-HB^1	4%	.37	.13	13%	.34	.14	36%	.32	.11	16%	.18	.05
Lapata, 2003	90%	.01	.04	58%	.02	.06	97%	.05	.04	46%	-.05	.00
	ACCIDENTS											
IDL-CH-A$^*_{}$	0%	.41	.21	0%	.40	.21	0%	.37	.15	0%	.13	.10
IDL-CH-HB^{100}	0%	.41	.20	0%	.40	.21	2%	.36	.15	0%	.12	.10
IDL-CH-HB^1	0%	.38	.19	12%	.32	.20	13%	.34	.13	33%	-.04	.06
Lapata, 2003	86%	.11	.03	67%	.12	.05	85%	.18	.00	24%	-.05	.06

Table 1: Evaluation of search algorithms for document coherence, for both EARTHQUAKES and ACCIDENTS genres, across the IBM_1^{D}, IBM_1^{I}, CM, and EB models. Performance is measured in terms of percentage of Estimated Search Errors (ESE), as well as quality of found realizations (average TAU and BLEU).

Model	TAU	BLEU	TAU	BLEU
	EARTHQUAKES		ACCIDENTS	
IBM_1^{D}	.38	.12	.41	.20
IBM_1^{I}	.32	.13	.40	.21
CM	.39	.12	.36	.15
EB	.19	.06	.12	.10
Log-linear$_{\text{uniform}}$	.34	.14	.48	.23
Log-linear$_{\text{maxTAU}}$	**.47**	.15	**.50**	.23
Log-linear$_{\text{maxBLEU}}$	.46	**.16**	.49	**.24**

Table 2: Evaluation of stochastic models for document coherence, for both EARTHQUAKES and ACCIDENTS genre, using IDL-CH-HB^{100}.

Overall performance	TAU	
	QUAKES	ACCID.
Lapata (2003)	0.48	0.07
Barzilay & Lee (2004)	**0.81**	0.44
Barzilay & Lee (reproduced)	0.39	0.36
Barzilay & Lapata (2005)	0.19	0.12
IBM_1^{D}, IDL-CH-HB^{100}	0.38	0.41
Log-lin$_{\text{maxTAU}}$, IDL-CH-HB^{100}	0.47	**0.50**

Table 3: Comparison of overall performance (affected by both model & search procedure) of our framework with previous results.

Board's database.

For both collections, we used 100 documents for training and 100 documents for testing. A fraction of 40% of the training documents was temporarily removed and used as a development set, on which we performed the discriminative training procedure.

4.2 Evaluation of Search Algorithms

We evaluated the performance of several search algorithms across four stochastic models of document coherence: the IBM_1^{D} and IBM_1^{I} coherence models, the content model of Barzilay and Lee (2004) (CM), and the entity-based model of Barzilay and Lapata (2005) (EB) (Section 2). We measure search performance using an Estimated Search Error (ESE) figure, which reports the percentage of times when the search algorithm proposes a sentence order which scores lower than the original sentence order (OSO). We also measure the quality of the proposed documents using TAU and BLEU, using as reference the OSO.

In Table 1, we report the performance of four search algorithms. The first three, IDL-CH-A*, IDL-CH-HB^{100}, and IDL-CH-HB^1 are the IDL-based search algorithms of Section 3, implementing A* search, histogram beam search with a beam of 100, and histogram beam search with a beam of 1, respectively. We compare our algorithms against the greedy algorithm used by Lapata (2003). We note here that the comparison is rendered meaningful by the observation that this algorithm performs search identically with algorithm IDL-CH-HB^1 (histogram beam 1), when setting the heuristic function for future costs a to constant 0.

The results in Table 1 clearly show the superiority of the IDL-CH-A* and IDL-CH-HB^{100} algorithms. Across all models considered, they consistently propose documents with scores at least as good as OSO (0% Estimated Search Error). As the original documents were coherent, it follows that the proposed document realizations also exhibit coherence. In contrast, the greedy algorithm of Lapata (2003) makes grave search errors. As the comparison between IDL-CH-HB^{100} and IDL-CH-HB^{1} shows, the superiority of the IDL-CH algorithms depends more on the admissible heuristic function a than in the ability to maintain multiple hypotheses while searching.

4.3 Evaluation of Log-linear Models

For this round of experiments, we held constant the search procedure (IDL-CH-HB^{100}), and varied the λ_m parameters of Equation 1. The utility-trained log-linear models are compared here against a baseline log-linear model log-linear$_{\text{uniform}}$, for which all λ_m parameters are set to 1, and also against the individual models. The results are presented in Table 2.

If not properly weighted, the log-linear combination may yield poorer results than those of individual models (average TAU of .34 for log-linear$_{\text{uniform}}$, versus .38 for IBM$_1^{\text{D}}$ and .39 for CM, on the EARTHQUAKES domain). The highest TAU accuracy is obtained when using TAU to perform utility-based training of the λ_m parameters (.47 for EARTHQUAKES, .50 for ACCIDENTS). The highest BLEU accuracy is obtained when using BLEU to perform utility-based training of the λ_m parameters (.16 for EARTHQUAKES, .24 for the ACCIDENTS). For both genres, the differences between the highest accuracy figures (in bold) and the accuracy of the individual models are statistically significant at 95% confidence (using bootstrap resampling).

4.4 Overall Performance Evaluation

The last comparison we provide is between the performance provided by our framework and previously-reported performance results (Table 3). We are able to provide this comparison based on the TAU figures reported in (Barzilay and Lee, 2004). The training and test data for both genres is the same, and therefore the figures can be directly compared. These figures account for combined model and search performance.

We first note that, unfortunately, we failed to accurately reproduce the model of Barzilay and Lee (2004). Our reproduction has an average TAU figure of only .39 versus the original figure of .81 for EARTHQUAKES, and .36 versus .44 for ACCIDENTS. On the other hand, we reproduced successfully the model of Barzilay and Lapata (2005), and the average TAU figure is .19 for EARTHQUAKES, and .12 for ACCIDENTS[3]. The large difference on the EARTHQUAKES corpus between the performance of Barzilay and Lee (2004) and our reproduction of their model is responsible for the overall lower performance (0.47) of our log-linear$_{\texttt{maxTAU}}$ model and IDL-CH-HB^{100} search algorithm, which is nevertheless higher than that of its component model CM (0.39). On the other hand, we achieve the highest accuracy figure (0.50) on the ACCIDENTS corpus, outperforming the previous-highest figure (0.44) of Barzilay and Lee (2004). These result empirically show that utility-trained log-linear models of discourse coherence outperform each of the individual coherence models considered.

5 Discussion and Conclusions

We presented a generic framework that is capable of integrating various stochastic models of discourse coherence into a more powerful model that combines the strengths of the individual models. An important ingredient of this framework are the search algorithms based on IDL-expressions, which provide a flexible way of solving discourse generation problems using stochastic models. Our generation algorithms are fundamentally different from previously-proposed algorithms for discourse generation. The genetic algorithms of Mellish et al. (1998) and Karamanis and Manarung (2002), as well as the greedy algorithm of Lapata (2003), provide no theoretical guarantees on the optimality of the solutions they propose. At the other end of the spectrum, the exhaustive search of Barzilay and Lee (2004), while ensuring optimal solutions, is prohibitively expensive, and cannot be used to perform utility-based training. The linear programming algorithm of Althaus et al. (2005) is the only proposal that achieves both good speed and accuracy. Their algorithm, however, cannot handle models with hidden states, cannot compute k-best lists, and does not have the representation flexibility provided by

[3] Note that these figures cannot be compared directly with the figures reported in (Barzilay and Lapata, 2005), as they use a different type of evaluation. Our EB model achieves the same performance as the original Syntax+Salience model, in their evaluation setting.

IDL-expressions, which is crucial for coherence decoding in realistic applications such as multi-document summarization.

For each of the coherence model combinations that we have utility trained, we obtained improved results on the discourse ordering problem compared to the individual models. This is important for two reasons. Our improvements can have an immediate impact on multi-document summarization applications (Barzilay et al., 2002). Also, our framework provides a solid foundation for subsequent research on discourse coherence models and related applications.

Acknowledgments This work was partially supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

References

Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. 2005. Computing locally coherent discourse. In *Proceedings of the ACL*, pages 399–406.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the ACL*, pages 141–148.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the HLT-NAACL*, pages 113–120.

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

L. Carlson, D. Marcu, and M. E. Okurowski. 2003. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In J. van Kuppevelt and R. Smith, eds., *Current Directions in Discourse and Dialogue*. Kluwer Academic Publishers.

K. Forbes, E. Miltsakaki, R. Prasad, A. Sarkar, A. Joshi, and B. Webber. 2001. D-LTAG System: Discourse parsing with a lexicalized tree-adjoining grammar. In *Workshop on Information Structure, Discourse Structure and Discourse Semantics*.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the International Workshop on Parsing Technologies (IWPT 2005)*.

Nikiforos Karamanis and Hisar M. Manurung. 2002. Stochastic text structuring using the principle of continuity. In *Proceedings of INLG*, pages 81–88.

Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. 2004. Evaluating centering-based metrics of coherence for text structuring using a reliably annotated corpus. In *Proc. of the ACL*.

Rodger Kibble and Richard Power. 2004. Optimising referential coherence in text generation. *Computational Linguistics*, 30(4):410–416.

Kevin Knight. 2003. Personal Communication.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with text ordering. In *Proceedings of the ACL*, pages 545–552.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Daniel Marcu. 1996. In *Proceedings of the Student Conference on Computational Linguistics*, pages 136-143.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press.

Chris Mellish, Alistair Knott, Jon Oberlander, and Mick O'Donnell. 1998. Experiments using stochastic search for text planning. In *Proceedings of the INLG*, pages 98–107.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Mark-Jan Nederhof and Giorgio Satta. 2004. IDL-expressions: a formalism for representing and parsing finite languages in natural language processing. *Journal of Artificial Intelligence Research*, pages 287–317.

Vincent Ng. 2005. Machine learning for coreference resolution: from local clasification to global reranking. In *Procedings of the ACL*, pages 157–164.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311–318.

Stuart Russell and Peter Norvig. 1995. *Artificial Intelligence. A Modern Approach*. Prentice Hall.

Donia R. Scott and Clarisse S. de Souza. 1990. Getting the message across in RST-based text generation. In Robert Dale, Chris Mellish, and Michael Zock, eds., *Current Research in Natural Language Generation*, pages 47–73. Academic Press.

Radu Soricut and Daniel Marcu. 2005. Towards developing generation algorithms for text-to-text applications. In *Proceedings of the ACL*, pages 66–74.

Radu Soricut. 2006. *Natural Language Generation for Text-to-Text Applications Using an Information-Slim Representation*. Ph.D. thesis, University of Southern California.