Probabilistic Text Structuring: Experiments with Sentence Ordering

Mirella Lapata
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield S1 4DP, UK
mlap@dcs.shef.ac.uk

Abstract

Ordering information is a critical task for natural language generation applications. In this paper we propose an approach to information ordering that is particularly suited for text-to-text generation. We describe a model that learns constraints on sentence order from a corpus of domain-specific texts and an algorithm that yields the most likely order among several alternatives. We evaluate the automatically generated orderings against authored texts from our corpus and against human subjects that are asked to mimic the model's task. We also assess the appropriateness of such a model for multidocument summarization.

1 Introduction

Structuring a set of facts into a coherent text is a non-trivial task which has received much attention in the area of concept-to-text generation (see Reiter and Dale 2000 for an overview). The structured text is typically assumed to be a tree (i.e., to have a hierarchical structure) whose leaves express the content being communicated and whose nodes specify how this content is grouped via rhetorical or discourse relations (e.g., contrast, sequence, elaboration).

For domains with large numbers of facts and rhetorical relations, there can be more than one possible tree representing the intended content. These different trees will be realized as texts with different sentence orders or even paragraph orders and different levels of coherence. Finding the tree that yields the best possible text is effectively a search problem. One way to address it is by narrowing down the search space either exhaustively or heuristically. Marcu (1997) argues that *global* coherence can be achieved if constraints on *local* coherence are satisfied. The latter are operationalized as weights on the ordering and adjacency of facts and are derived from a corpus of naturally occurring texts. A constraint satisfaction algorithm is used to find the tree with maximal weights from the space of all possible trees. Mellish et al. (1998) advocate stochastic search as an alternative to exhaustively examining the search space. Rather than requiring a global optimum to be found, they use a genetic algorithm to select a tree that is coherent enough for people to understand (local optimum).

The problem of finding an acceptable ordering does not arise solely in concept-to-text generation but also in the emerging field of text-to-text generation (Barzilay, 2003). Examples of applications that require some form of text structuring, are single- and multidocument summarization as well as question answering. Note that these applications do not typically assume rich semantic knowledge organized in tree-like structures or communicative goals as is often the case in concept-to-text generation. Although in single document summarization the position of a sentence in a document can provide cues with respect to its ordering in the summary, this is not the case in multidocument summarization where sentences are selected from different documents and must be somehow ordered so as to produce a coherent summary (Barzilay et al., 2002). Answering a question may also involve the extraction, potentially summarization, and ordering of information across multiple information sources.

Barzilay et al. (2002) address the problem of information ordering in multidocument summarization and show that naive ordering algorithms such as majority ordering (selects most frequent orders across input documents) and chronological ordering (orders facts according to publication date) do not always yield coherent summaries although the latter produces good results when the information is event-based. Barzilay et al. further conduct a study where subjects are asked to produce a coherent text from the output of a multidocument summarizer. Their results reveal that although the generated orders differ from subject to subject, topically related sentences always appear together. Based on the human study they propose an algorithm that first identifies topically related groups of sentences and then orders them according to chronological information.

In this paper we introduce an unsupervised probabilistic model for text structuring that learns ordering constraints from a large corpus. The model operates on sentences rather than facts in a knowledge base and is potentially useful for text-to-text generation applications. For example, it can be used to order the sentences obtained from a multidocument summarizer or a question answering system. Sentences are represented by a set of informative features (e.g., a verb and its subject, a noun and its modifier) that can be automatically extracted from the corpus without recourse to manual annotation.

The model learns which sequences of features are likely to co-occur and makes predictions concerning preferred orderings. Local coherence is thus operationalized by sentence proximity in the training corpus. Global coherence is obtained by greedily searching through the space of possible orders. As in the case of Mellish et al. (1998) we construct an acceptable ordering rather than the best possible one. We propose an automatic method of evaluating the orders generated by our model by measuring closeness or distance from the gold standard, a collection of orders produced by humans.

The remainder of this paper is organized as follows. Section 2 introduces our model and an algorithm for producing a possible order. Section 3 describes our corpus and the estimation of the model parameters. Our experiments are detailed in Section 4. We conclude with a discussion in Section 5.

2 Learning to Order

Given a collection of texts from a particular domain, our task is to learn constraints on the ordering of their sentences. In the training phase our model will learn these constraints from adjacent sentences represented by a set of informative features. In the testing phase, given a set of unseen sentences, we will rely on our prior experience of how sentences are usually ordered for choosing the most likely ordering.

2.1 The Model

We express the probability of a text made up of sentences $S_1 \dots S_n$ as shown in (1). According to (1), the task of predicting the next sentence is dependent on its $n-i$ previous sentences.

$$(1)\quad \begin{aligned} P(T) &= P(S_1 \dots S_n) \\ &= P(S_1)P(S_2|S_1)P(S_3|S_1,S_2)\dots P(S_n|S_1 \dots S_{n-1}) \\ &= \prod_{i=1}^{n} P(S_n|S_1 \dots S_{n-i}) \end{aligned}$$

We will simplify (1) by assuming that the probability of any given sentence is determined only by its previous sentence:

$$(2)\quad \begin{aligned} P(T) &= P(S_1)P(S_2|S_1)P(S_3|S_2)\dots P(S_n|S_{n-1}) \\ &= \prod_{i=1}^{n} P(S_i|S_{i-1}) \end{aligned}$$

This is a somewhat simplistic attempt at capturing Marcu's (1997) local coherence constraints as well as Barzilay et al.'s (2002) observations about topical relatedness. While this is clearly a naive view of text coherence, our model has some notion of the types of sentences that typically go together, even though it is agnostic about the specific rhetorical relations that glue sentences into a coherent text. Also note that the simplification in (2) will make the estimation of the probabilities $P(S_i|S_{i-1})$ more reliable in the face of sparse data. Of course estimating $P(S_i|S_{i-1})$ would be impossible if S_i and S_{i-1} were actual sentences. It is unlikely to find the exact same sentence repeated several times in a corpus. What we can find and count is the number of times a given structure or word appears in the corpus. We will therefore estimate $P(S_i|S_{i-1})$ from features that express its structure and content (these features are described in detail in Section 3):

$$(3)\quad \begin{aligned} &P(S_i|S_{i-1}) = \\ &P(\langle a_{\langle i,1\rangle}, a_{\langle i,2\rangle} \dots a_{\langle i,n\rangle}\rangle | \langle a_{\langle i-1,1\rangle}, a_{\langle i-1,2\rangle} \dots a_{\langle i-1,m\rangle}\rangle) \end{aligned}$$

where $\langle a_{\langle i,1\rangle}, a_{\langle i,2\rangle} \dots a_{\langle i,n\rangle}\rangle$ are features relevant for sentence S_i and $\langle a_{\langle i-1,1\rangle}, a_{\langle i-1,2\rangle} \dots a_{\langle i-1,m\rangle}\rangle$ for sentence S_{i-1}. We will assume that these features are independent and that $P(S_i|S_{i-1})$ can be estimated from the pairs in the Cartesian product defined over the features expressing sentences S_i and S_{i-1}: $(a_{\langle i,j\rangle}, a_{\langle i-1,k\rangle}) \in S_i \times S_{i-1}$. Under these assumptions $P(S_i|S_{i-1})$ can be written as follows:

$$(4)\quad \begin{aligned} P(S_i|S_{i-1}) &= P(a_{\langle i,1\rangle}|a_{\langle i-1,1\rangle})\dots P(a_{\langle i,n\rangle}|a_{\langle i-1,m\rangle}) \\ &= \prod_{(a_{\langle i,j\rangle}, a_{\langle i-1,k\rangle}) \in S_i \times S_{i-1}} P(a_{\langle i,j\rangle}|a_{\langle i-1,k\rangle}) \end{aligned}$$

Assuming that the features are independent again makes parameter estimation easier. The Cartesian product over the features in S_i and S_{i-1} is an attempt to capture inter-sentential dependencies. Since

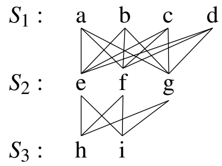

Figure 1: Example of probability estimation

we don't know a priori what the important feature combinations are, we are considering all possible combinations over two sentences. This will admittedly introduce some noise, given that some dependencies will be spurious, but the model can be easily retrained for different domains for which different feature combinations will be important. The probability $P(a_{\langle i,j\rangle}|a_{\langle i-1,k\rangle})$ is estimated as:

$$(5) \quad P(a_{\langle i,j\rangle}|a_{\langle i-1,k\rangle}) = \frac{f(a_{\langle i,j\rangle},a_{\langle i-1,k\rangle})}{\sum\limits_{a_{\langle i,j\rangle}} f(a_{\langle i,j\rangle},a_{\langle i-1,k\rangle})}$$

where $f(a_{\langle i,j\rangle},a_{\langle i-1,k\rangle})$ is the number of times feature $a_{\langle i,j\rangle}$ is preceded by feature $a_{\langle i-1,k\rangle}$ in the corpus. The denominator expresses the number of times $a_{\langle i-1,k\rangle}$ is attested in the corpus (preceded by any feature). The probabilities $P(a_{\langle i,j\rangle}|a_{\langle i-1,k\rangle})$ will be unreliable when the frequency estimates for $f(a_{\langle i,j\rangle},a_{\langle i-1,k\rangle})$ are small, and undefined in cases where the feature combinations are unattested in the corpus. We therefore smooth the observed frequencies using back-off smoothing (Katz, 1987).

To illustrate with an example consider the text in Figure 1 which has three sentences S_1, S_2, S_3, each represented by their respective features denoted by letters. The probability $P(S_3|S_2)$ will be calculated by taking the product of $P(h|e)$, $P(h|f)$, $P(h|g)$, $P(i|e)$, $P(i|f)$, and $P(i|g)$. To obtain $P(h|e)$, we need $f(h,e)$ and $f(e)$ which can be estimated in Figure 1 by counting the number of edges connecting e and h and the number of edges starting from e, respectively. So, $P(h|e)$ will be 0.16 given that $f(h,e)$ is one and $f(e)$ is six (see the normalization in (5)).

2.2 Determining an Order

Once we have collected the counts for our features we can determine the order for a new text that we haven't encountered before, since some of the features representing its sentences will be familiar. Given a text with N sentences there are $N!$ possible orders. The set of orders can be represented as a complete graph, where the set of vertices V is equal to the set of sentences S and each edge $u \rightarrow v$ has a weight, the probability $P(u|v)$. Cohen et al. (1999) show that the problem of finding an optimal ordering through a directed weighted graph is NP-complete. Fortunately, they propose a simple greedy algorithm that provides an approximate solution which can be easily modified for our task (see also Barzilay et al. 2002).

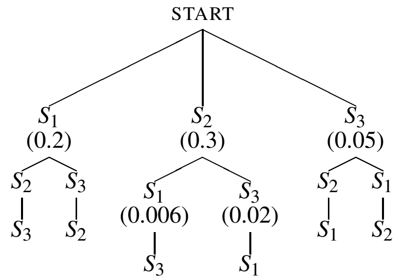

Figure 2: Finding an order for a three sentence text

The algorithm starts by assigning each vertex $v \in V$ a probability. Recall that in our case vertices are sentences and their probabilities can be calculated by taking the product of the probabilities of their features. The greedy algorithm then picks the node with the highest probability and orders it ahead of the other nodes. The selected node and its incident edges are deleted from the graph. Each remaining node is now assigned the conditional probability of seeing this node given the previously selected node (see (4)). The node which yields the highest conditional probability is selected and ordered ahead. The process is repeated until the graph is empty.

As an example consider again a three sentence text. We illustrate the search for a path through the graph in Figure 2. First we calculate which of the three sentences S_1, S_2, and S_3 is most likely to start the text (during training we record which sentences appear in the beginning of each text). Assuming that $P(S_2|\text{START})$ is the highest, we will order S_2 first, and ignore the nodes headed by S_1 and S_3. We next compare the probabilities $P(S_1|S_2)$ and $P(S_3|S_2)$. Since $P(S_3|S_2)$ is more likely than $P(S_1|S_2)$, we order S_3 after S_2 and stop, returning the order S_2, S_3, and S_1. As can be seen in Figure 2 for each vertex we keep track of the most probable edge that ends in that vertex, thus setting th beam search width to one.

Note, that equation (4) would assign lower and lower probabilities to sentences with large numbers of features. Since we need to compare sentence pairs with varied numbers of features, we will normalize the conditional probabilities $P(S_i|S_{i-1})$ by the number feature of pairs that form the Cartesian product over S_i and S_{i-1}.

1. Laidlaw Transportation Ltd. said shareholders will be asked at its Dec. 7 annual meeting to approve a change of name to Laidlaw Inc.
2. The company said its existing name hasn't represented its businesses since the 1984 sale of its trucking operations.
3. Laidlaw is a waste management and school-bus operator, in which Canadian Pacific Ltd. has a 47% voting interest.

Figure 3: A text from the BLLIP corpus

3 Parameter Estimation

The model in Section 2.1 was trained on the BLLIP corpus (30 M words), a collection of texts from the Wall Street Journal (years 1987-89). The corpus contains 98,732 stories. The average story length is 19.2 sentences. 71.30% of the texts in the corpus are less than 50 sentences long. An example of the texts in this newswire corpus is shown in Figure 3.

The corpus is distributed in a Treebank-style machine-parsed version which was produced with Charniak's (2000) parser. The parser is a "maximum-entropy inspired" probabilistic generative model. It achieves 90.1% average precision/recall for sentences with maximum length 40 and 89.5% for sentences with maximum length 100 when trained and tested on the standard sections of the Wall Street Journal Treebank (Marcus et al., 1993).

We also obtained a dependency-style version of the corpus using MINIPAR (Lin, 1998) a broad coverage parser for English which employs a manually constructed grammar and a lexicon derived from WordNet with an additional dictionary of proper names (130,000 entries in total). The grammar is represented as a network of 35 nodes (i.e., grammatical categories) and 59 edges (i.e., types of syntactic (dependency) relations). The output of MINIPAR is a dependency graph which represents the dependency relations between words in a sentence (see Table 1 for an example). Lin (1998) evaluated the parser on the SUSANNE corpus (Sampson, 1996), a domain independent corpus of British English, and achieved a recall of 79% and precision of 89% on the dependency relations.

From the two different parsed versions of the BLLIP corpus the following features were extracted:

Verbs. Investigations into the interpretation of narrative discourse (Asher and Lascarides, 2003) have shown that specific lexical information (e.g., verbs, adjectives) plays an important role in determining the discourse relations between propositions. Although we don't have an explicit model of rhetorical relations and their effects on sentence ordering, we capture the lexical inter-dependencies between sentences by focusing on verbs and their precedence relationships in the corpus.

From the Treebank parses we extracted the verbs contained in each sentence. We obtained two versions of this feature: (a) a lemmatized version where verbs were reduced to their base forms and (b) a non-lemmatized version which preserved tense-related information; more specifically, verbal complexes (e.g., *I will have been going*) were identified from the parse trees heuristically by devising a set of 30 patterns that search for sequences of modals, auxiliaries and verbs. This is an attempt at capturing temporal coherence by encoding sequences of events and their morphology which indirectly indicates their tense.

To give an example consider the text in Figure 3. For the lemmatized version, sentence (1) will be represented by *say*, *will*, *be*, *ask*, and *approve*; for the tensed version, the relevant features will be *said*, *will be asked*, and *to approve*.

Nouns. Centering Theory (CT, Grosz et al. 1995) is an entity-based theory of local coherence, which claims that certain entities mentioned in an utterance are more central than others and that this property constrains a speaker's use of certain referring expressions. The principles underlying CT (e.g., continuity, salience) are of interest to concept-to-text generation as they offer an entity-based model of text and sentence planning which is particularly suited for descriptional genres (Kibble and Power, 2000).

We operationalize entity-based coherence for text-to-text generation by simply keeping track of the nouns attested in a sentence without however taking personal pronouns into account. This simplification is reasonable if one has text-to-text generation mind. In multidocument summarization for example, sentences are extracted from different documents; the referents of the pronouns attested in these sentences are typically not known and in some cases identical pronouns may refer to different entities. So making use of noun-pronoun or pronoun-pronoun co-occurrences will be uninformative or in fact misleading.

We extracted nouns from a lemmatized version of the Treebank-style parsed corpus. In cases of noun compounds, only the compound head (i.e., rightmost noun) was taken into account. A small set of rules was used to identify organizations (e.g., *United Laboratories Inc.*), person names (e.g., *Jose Y. Campos*), and locations (e.g., *New England*) spanning more than one word. These were grouped together and were also given the general categories `person`, `organization`, and `location`. The model backs off to these categories when unknown person names, locations, and organizations are encountered. Dates, years, months and numbers were substituted by the categories `date`, `year`, `month`, and `number`.

In sentence (1) (see Figure 3) we identify the nouns *Laidlaw Transportation Ltd.*, *shareholder*, *Dec 7*, *meeting*, *change*, *name* and *Laidlaw Inc.* In sentence (2) the relevant nouns are *company*, *name*, *business*, *1984*, *sale*, and *operation*.

Dependencies. Note that the noun and verb features do not capture the structure of the sentences to be ordered. This is important for our domain, as texts seem to be rather formulaic and similar syntactic structures are often used (e.g., direct and indirect speech, restrictive relative clauses, predicative structures). In this domain companies typically say things, and texts often begin with a statement of what a company or an individual has said (see sentence (1) in Figure 3). Furthermore, companies and individuals are described with certain attributes (persons can be presidents or governors, companies are bankrupt or manufacturers, etc.) that can give clues for inferring coherence.

The dependencies were obtained from the output of MINIPAR. Some of the dependencies for sentence (2) from Figure 3 are shown in Table 1. The dependencies capture structural as well lexical information. They are represented as triples, consisting of a head (leftmost element, e.g., *say*, *name*), a modifier (rightmost element, e.g., *company*, *its*) and a relation (e.g., subject (`V:subj:N`), object (`V:obj:N`), modifier (`N:mod:A`)).

For efficiency reasons we focused on triples whose dependency relations (e.g., `V:subj:N`) were attested in the corpus with frequency larger than one per million. We further looked at how individual types of relations contribute to the ordering task. More specifically we experimented with dependencies relating to verbs (49 types), nouns (52 types), verbs and nouns (101 types) (see Table 1 for examples). We also ran a version of our model with all types of relations, including adjectives, adverbs and prepositions (147 types in total).

Verb	Noun
say `V:subj:N` company	name `N:gen:N` its
represent `V:subj:N` name	name `N:mod:A` existing
represent `V:have:have` have	business `N:gen:N` its
represent `V:obj:N` business	business `N:mod:Prep` since
	company `N:det:Det` the

Table 1: Dependencies for sentence (2) in Figure 3

	A	B	C	D	E	F	G	H	I	J
Model 1	1	2	3	4	5	6	7	8	9	10
Model 2	2	1	5	3	4	6	7	9	8	10
Model 3	10	2	3	4	5	6	7	8	9	1

Table 2: Example of rankings for a 10 sentence text

4 Experiments

In this section we describe our experiments with the model and the features introduced in the previous sections. We first evaluate the model by attempting to reproduce the structure of unseen texts from the BLLIP corpus, i.e., the corpus on which the model is trained on. We next obtain an upper bound for the task by conducting a sentence ordering experiment with humans and comparing the model against the human data. Finally, we assess whether this model can be used for multi-document summarization using data from Barzilay et al. (2002). But before we outline the details of our experiments we discuss our choice of metric for comparing different orders.

4.1 Evaluation Metric

Our task is to produce an ordering for the sentences of a given text. We can think of the sentences as objects for which a ranking must be produced. Table 2 gives an example of a text containing 10 sentences (A–J) and the orders (i.e., rankings) produced by three hypothetical models.

A number of metrics can be used to measure the distance between two rankings such as Spearman's correlation coefficient for ranked data, Cayley distance, or Kendall's τ (see Lebanon and Lafferty 2002 for details). Kendall's τ is based on the number of *inversions* in the rankings and is defined in (6):

(6) $$\tau = 1 - \frac{2(\text{number of inversions})}{N(N-1)/2}$$

where N is the number of objects (i.e., sentences) being ranked and inversions are the number of interchanges of consecutive elements necessary to arrange them in their natural order. If we think in terms of permutations, then τ can be interpreted as the minimum number of adjacent transpositions needed to bring one order to the other. In Table 2 the number of inversions can be calculated by counting the number of intersections of the lines. The metric ranges from -1 (inverse ranks) to 1 (identical ranks). The τ for Model 1 and Model 2 in Table 2 is .822.

Kendall's τ seems particularly appropriate for the tasks considered in this paper. The metric is sensitive to the fact that some sentences may be always ordered next to each other even though their absolute orders might differ. It also penalizes inverse rankings. Comparison between Model 1 and Model 3 would give a τ of 0.244 even though the orders between the two models are identical modulo the beginning and the end. This seems appropriate given that flipping the introduction in a document with the conclusions seriously disrupts coherence.

4.2 Experiment 1: Ordering Newswire Texts

The model from Section 2.1 was trained on the BLLIP corpus and tested on 20 held-out randomly selected unseen texts (average length 15.3). We also used 20 randomly chosen texts (disjoint from the test data) for development purposes (average length 16.2). All our results are reported on the test set.

The input to the the greedy algorithm (see Section 2.2) was a text with a randomized sentence ordering. The ordered output was compared against the original authored text using τ. Table 3 gives the average τ (T) for all 20 test texts when the following features are used: lemmatized verbs (V_L), tensed verbs (V_T), lemmatized nouns (N_L), lemmatized verbs and nouns (V_LN_L), tensed verbs and lemmatized nouns (V_TN_L), verb-related dependencies (V_D), noun-related dependencies (N_D), verb and noun dependencies (V_DN_D), and all available dependencies (A_D). For comparison we also report the naive baseline of generating a random oder (B_R). As can be seen from Table 3 the best performing features are N_L and V_DN_D. This is not surprising given that N_L encapsulates notions of entity-based coherence, which is relatively important for our domain. A lot of texts are about a particular entity (company or individual) and their properties. The feature V_DN_D subsumes several other features and does expectedly better: it captures entity-based coherence, the interrelations among verbs, the structure of sentences and also preserves information about argument structure (who is doing what to whom). The distance between the orders produced by the model and the original texts increases when all types of dependencies are taken into account. The feature space becomes too big, there are too many spurious feature pairs, and the model can't distinguish informative from non-informative features.

Feature	T	StdDev	Min	Max
B_R	.35	.09	.17	.47
V_L	.44	.24	.17	.93
V_T	.46	.21	.17	.80
N_L	**.54**	.16	.18	.76
V_LN_L	.46	.12	.18	.61
V_TN_L	.49	.17	.21	.86
V_D	.51	.17	.10	.83
N_D	.45	.17	.10	.67
V_DN_D	**.57**	.12	.62	.83
A_D	.48	.17	.10	.83

Table 3: Comparison between original BLLIP texts and model generated variants

We carried out a one-way Analysis of Variance (ANOVA) to examine the effect of different feature types. The ANOVA revealed a reliable effect of feature type ($F(9,171) = 3.31$; $p < 0.01$). We performed Post-hoc Tukey tests to further examine whether there are any significant differences among the different features and between our model and the baseline. We found out that N_L, V_TN_L, V_D, and V_DN_D are significantly better than B_R ($\alpha = 0.01$), whereas N_L and V_DN_D are not significantly different from each other. However, they are significantly better than all other features ($\alpha = 0.05$).

4.3 Experiment 2: Human Evaluation

In this experiment we compare our model's performance against human judges. Twelve texts were randomly selected from the 20 texts in our test data. The texts were presented to subjects with the order of their sentences scrambled. Participants were asked to reorder the sentences so as to produce a coherent text. Each participant saw three texts randomly chosen from the pool of 12 texts. A random order of sentences was generated for every text the participants saw. Sentences were presented verbatim, pronouns and connectives were retained in order to make ordering feasible. Notice that this information is absent from the features the model takes into account.

The study was conducted remotely over the Internet using a variant of Barzilay et al.'s (2002) software. Subjects first saw a set of instructions that explained the task, and had to fill in a short questionnaire including basic demographic information. The experiment was completed by 137 volunteers (approximately 33 per text), all native speakers of English. Subjects were recruited via postings to local

Feature	T	StdDev	Min	Max
V_L	.45	.16	.10	.90
V_T	.46	.18	.10	.90
N_L	**.51**	.14	.10	.90
$V_L N_L$	.44	.14	.18	.61
$V_T N_L$	.49	.18	.21	.86
V_D	.47	.14	.10	.93
N_D	.46	.15	.10	.86
$V_D N_D$	**.55**	.15	.10	.90
A_D	.48	.16	.10	.83
H_H	**.58**	.08	.26	.75

Table 4: Comparison between orderings produced by humans and the model on BLLIP texts

Features	T	StdDev	Min	Max
B_R	.43	.13	.19	.97
N_L	.48	.16	.21	.86
$V_D N_D$	**.56**	.13	.32	.86
H_H	**.60**	.17	-1	.98

Table 5: Comparison between orderings produced by humans and the model on multidocument summaries

Usenet newsgroups.

Table 4 reports pairwise τ averaged over 12 texts for all participants (H_H) and the average τ between the model and each of the subjects for all features used in Experiment 1. The average distance in the orderings produced by our subjects is .58. The distance between the humans and the best features is .51 for N_L and .55 for $V_D N_D$. An ANOVA yielded a significant effect of feature type ($F(9,99) = 5.213$; $p < 0.01$). Post-hoc Tukey tests revealed that V_L, V_T, V_D, N_D, A_D, $V_L N_L$, and $V_T N_L$ perform significantly worse than H_H ($\alpha = 0.01$), whereas N_L and $V_D N_D$ are not significantly different from H_H ($\alpha = 0.01$). This is in agreement with Experiment 1 and points to the importance of lexical and structural information for the ordering task.

4.4 Experiment 3: Summarization

Barzilay et al. (2002) collected a corpus of multiple orderings in order to study what makes an order cohesive. Their goal was to improve the ordering strategy of MULTIGEN (McKeown et al., 1999) a multidocument summarization system that operates on news articles describing the same event. MULTIGEN identifies text units that convey similar information across documents and clusters them into themes. Each theme is next syntactically analysed into predicate argument structures; the structures that are repeated often enough are chosen to be included into the summary. A language generation system outputs a sentence (per theme) from the selected predicate argument structures.

Barzilay et al. (2002) collected ten sets of articles each consisting of two to three articles reporting the same event and simulated MULTIGEN by manually selecting the sentences to be included in the final summary. This way they ensured that orderings were not influenced by mistakes their system could have made. Explicit references and connectives were removed from the sentences so as not to reveal clues about the sentence ordering. Ten subjects provided orders for each summary which had an average length of 8.8.

We simulated the participants' task by using the model from Section 2.1 to produce an order for each candidate summary[1]. We then compared the differences in the orderings generated by the model and participants using the best performing features from Experiment 2 (i.e., N_L and $V_D N_D$). Note that the model was trained on the BLLIP corpus, whereas the sentences to be ordered were taken from news articles describing the same event. Not only were the news articles unseen but also their syntactic structure was unfamiliar to the model. The results are shown in table 5, again average pairwise τ is reported. We also give the naive baseline of choosing a random order (B_R). The average distance in the orderings produced by Barzilay et al.'s (2002) participants is .60. The distance between the humans and N_L is .48 whereas the average distance between $V_D N_D$ and the humans is .56. An ANOVA yielded a significant effect of feature type ($F(3,27) = 15.25$; $p < 0.01$). Post-hoc Tukey tests showed that $V_D N_D$ was significantly better than B_R, but N_L wasn't. The difference between $V_D N_D$ and H_H was not significant.

Although N_L performed adequately in Experiments 1 and 2, it failed to outperform the baseline in the summarization task. This may be due to the fact that entity-based coherence is not as important as temporal coherence for the news articles summaries. Recall that the summaries describe events across documents. This information is captured more adequately by $V_D N_D$ and not by N_L that only keeps a record of the entities in the sentence.

5 Discussion

In this paper we proposed a data intensive approach to text coherence where constraints on sentence ordering are learned from a corpus of domain-specific

[1] The summaries as well as the human data are available from http://www.cs.columbia.edu/~noemie/ordering/.

texts. We experimented with different feature encodings and showed that lexical and syntactic information is important for the ordering task. Our results indicate that the model can successfully generate orders for texts taken from the corpus on which it is trained. The model also compares favorably with human performance on a single- and multiple document ordering task.

Our model operates on the surface level rather than the logical form and is therefore suitable for text-to-text generation systems; it acquires ordering constraints automatically, and can be easily ported to different domains and text genres. The model is particularly relevant for multidocument summarization since it could provide an alternative to chronological ordering especially for documents where publication date information is unavailable or uninformative (e.g., all documents have the same date). We proposed Kendall's τ as an automated method for evaluating the generated orders.

There are a number of issues that must be addressed in future work. So far our evaluation metric measures order similarities or dissimilarities. This enables us to assess the importance of particular feature combinations automatically and to evaluate whether the model and the search algorithm generate potentially acceptable orders without having to run comprehension experiments each time. Such experiments however are crucial for determining how coherent the generated texts are and whether they convey the same semantic content as the originally authored texts. For multidocument summarization comparisons between our model and alternative ordering strategies are important if we want to pursue this approach further.

Several improvements can take place with respect to the model. An obvious question is whether a trigram model performs better than the model presented here. The greedy algorithm implements a search procedure with a beam of width one. In the future we plan to experiment with larger widths (e.g., two or three) and also take into account features that express semantic similarities across documents either by relying on WordNet or on automatic clustering methods.

Acknowledgments

The author was supported by EPSRC grant number R40036. We are grateful to Regina Barzilay and Noemie Elhadad for making available their software and for providing valuable comments on this work. Thanks also to Stephen Clark, Nikiforos Karamanis, Frank Keller, Alex Lascarides, Katja Markert, and Miles Osborne for helpful comments and suggestions.

References

Asher, Nicholas and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Barzilay, Regina. 2003. *Information Fusion for Multi-Document Summarization: Praphrasing and Generation*. Ph.D. thesis, Columbia University.

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

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*. Seattle, WA, pages 132–139.

Cohen, William W., Robert E. Schapire, and Yoram Singer. 1999. Learning to order things. *Journal of Artificial Intelligence Research* 10:243–270.

Grosz, Barbara, Aravind Joshi, , and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics* 21(2):203–225.

Katz, Slava M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics Speech and Signal Processing* 33(3):400–401.

Kibble, Rodger and Richard Power. 2000. An integrated framework for text planning and pronominalisation. In *In Proceedings of the 1st International Conference on Natural Language Generation*. Mitzpe Ramon, Israel, pages 77–84.

Lebanon, Guy and John Lafferty. 2002. Combining rankings using conditional probability models on permutations. In C. Sammut and A. Hoffmann, editors, *In Proceedings of the 19th International Conference on Machine Learning*. Morgan Kaufmann Publishers, San Francisco, CA.

Lin, Dekang. 1998. Dependency-based evaluation of MINIPAR. In *In Proceedings on of the LREC Workshop on the Evaluation of Parsing Systems*. Granada, pages 48–56.

Marcu, Daniel. 1997. From local to global coherence: A bottom-up approach to text planning. In *In Proceedings of the 14th National Conference on Artificial Intelligence*. Providence, Rhode Island, pages 629–635.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* 19(2):313–330.

McKeown, Kathleen R., Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the 16th National Conference on Artificial Intelligence*. Orlando, FL, pages 453–459.

Mellish, Chris, Alistair Knott, Jon Oberlander, and Mick O' Donnell. 1998. Experiments using stochastic search for text planning. In *In Proceedings of the 9th International Workshop on Natural Language Generation*. Ontario, Canada, pages 98–107.

Reiter, Ehud and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

Sampson, Geoffrey. 1996. *English for the Computer*. Oxford University Press.