Database-Text Alignment via Structured Multilabel Classification

Benjamin Snyder and Regina Barzilay
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{bsnyder,regina}@csail.mit.edu

Abstract

This paper addresses the task of aligning a database with a corresponding text. The goal is to link individual database entries with sentences that verbalize the same information. By providing explicit semantics-to-text links, these alignments can aid the training of natural language generation and information extraction systems. Beyond these pragmatic benefits, the alignment problem is appealing from a modeling perspective: the mappings between database entries and text sentences exhibit rich structural dependencies, unique to this task. Thus, the key challenge is to make use of as many global dependencies as possible without sacrificing tractability. To this end, we cast text-database alignment as a structured multilabel classification task where each sentence is labeled with a subset of matching database entries. In contrast to existing multilabel classifiers, our approach operates over arbitrary global features of inputs and proposed labels. We compare our model with a baseline classifier that makes locally optimal decisions. Our results show that the proposed model yields a 15% relative reduction in error, and compares favorably with human performance.

1 Introduction

Uncovering the mapping between text and its underlying semantic representation lies at the core of natural language processing. For instance, the goal of information extraction is to structurally represent the semantic content of a given text. Natural language generation, on the other hand, seeks to create such texts from structured, non-linguistic representations. Training both information extraction and natural language generation systems generally requires annotated data that specifies the mapping between structured representations and corresponding fragments of text. Creating these mappings by hand is prohibitively expensive.

An alternative source of data for learning text-to-semantics mappings are parallel collections of databases and their corresponding texts. Such parallel collections are abundant and are readily available in multiple domains, including terrorism, sports, finance, and weather. However, text-database pairs cannot be used directly for system training without more refined alignment of their components. Such an alignment should explicitly link database entries to the sentences that verbalize their content.

In this paper, we investigate a supervised learning approach to aligning database entries and sentences. For example, consider a database containing statistics on an American football game and the corresponding newspaper summary. Figure 1 shows an excerpt from such a database and several summary sentences along with the target alignment.

Text-database alignment differs in several important respects from word alignment in machine translation. First, the databases and texts are not exactly parallel: many database entries may not be verbalized in a text, and conversely, some sentences may contain information not found in a database. Second, the number of items to be aligned is greater than the number of words in translated sentences. A typical database may contain hundreds of entries compared to an average of 25 words in a newspaper sentence. Finally, the database and text units to be aligned exhibit rich internal structure and complex interdependencies. Thus, the database schema provides a key for mining semantic relations between entries and corresponding sentences. For instance, in Figure 1 we can see that the three entries aligned to the final sentence bear a strong relationship to one another: the interception leads to a drive which culminates in a touchdown run. This relationship and many others like it can be determined by examining the database structure.

One possible approach is to formulate semantics-to-text alignment as simple binary classification of sentence-entry pairs. By considering the content overlap between a database entry and a sentence, the classifier determines whether they are aligned. This approach, however, fails to capture dependencies between local alignment decisions such as the one we just saw. Local alignments could also lead to overmatching when a sentence contains a single anchor that locally matches multiple database entries. For instance, the final sentence in Figure 1 contains the name "Brown" and the number "2." Besides matching an entry in the *play-by-play* table, these anchors also match the second entry in the *fumbles* table.[1] By making independent decisions for each of these entries, a lo-

[1] In American Football, a "fumble" occurs when a ball has been dropped and may be recovered by either team.

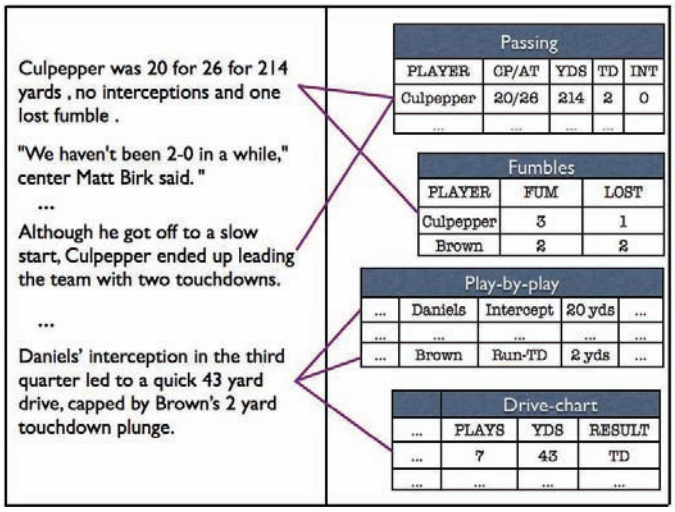

Figure 1: Sample target alignment between the NFL database and summary text.

cal classifier may erroneously align them both to the same sentence.

To capture global features of alignments, we cast our problem as a structured multilabel classification task. In the multilabel framework, each instance can be assigned *any subset* of the labels. In our case, we treat the sentences as instances and label them with the subset of database entries that match in content. By jointly considering a sentence with an entire set of candidate entries, our method ensures the global coherence of the aligned set. Moreover, we guide the alignment towards sets of entries with favorable structural properties. This task is accomplished by using a model that operates over arbitrary global features of the sentence and proposed entry set. However, the number of potential entry sets per sentence is exponential, and therefore this feature space cannot be searched exhaustively. To focus on promising label sets, we first rank all the database entries based on their individual match to the input sentence. We then only consider entry sets formed by choosing a cut-off in this ranking. Our model then selects among the remaining sets by examining their global alignment features. Although the target entry set may not be among these candidates, we train our model to choose the set closest to the target in Hamming distance.

We evaluate our algorithm on a corpus of manually aligned text-database pairs in the sports domain. To assess the contribution of our global model, we compare our method with a local classifier which treats every sentence-entry pair independently. Our method achieves a 15% relative reduction in error over this baseline, demonstrating the contribution of global features in our task.

2 Related Work

Multilabel classification A typical multilabel classifier has two stages: first, the labels of each input are ranked by a local scoring function. Then, a regression model predicts a threshold as a function of the label ranking scores. Finally, all the labels above the predicted threshold are returned. Research has focused on developing local ranking models aiming to minimize the number of incorrect labels predicted above correct ones, often ignoring threshold selection or treating it as a secondary concern [Elisseeff and Weston, 2001; Crammer and Singer, 2002; McDonald *et al.*, 2005; Schapire and Singer, 1999].

Our approach shares the initial local ranking step with traditional multilabel classifiers. However, in a task with structured labels, where the set of correct labels exhibit compatibility properties, it is insufficient to choose a cutoff by merely looking at the local ranking scores. Therefore, we treat ranking as a mere *pruning* step and learn to make intelligent threshold selections based on ranking scores as well as relations between entries. By considering collective properties of various label sets, we can mitigate ranking errors.

Ghamrawi and McCallum [2005] propose an approach to multilabel classification using Conditional Random Fields. Their approach models co-occurrence dependencies between pairs of class labels and individual input features. Our approach, on the other hand, learns weights for features defined *jointly* on the input and sets of (structured) labels. Also, our global feature function is not constrained to examining *pairs* of labels, but instead operates over arbitrary label sets.

Alignment The problem of word alignment of parallel bilingual corpora has been extensively studied in machine translation. While most of the research focuses on unsupervised models [Brown *et al.*, 1993], recently a number of supervised discriminative approaches have been proposed [Taskar *et al.*, 2005; Lacoste-Julien *et al.*, 2006]. Taskar *et al.* [2005] cast alignment as a graph matching problem and solve it in an optimization framework. Their model finds the global alignment which maximizes word-pair alignment scores, subject to a hard fertility constraint. Recently, they have reformulated the optimization problem to instead apply *soft* constraints on alignment fertility as well as control for other properties specific to word alignment [Lacoste-Julien *et al.*, 2006].

Our model differs in that it can consider arbitrary features of the alignment decisions. For instance, depending on the structural properties of the aligned objects, we often wish to *encourage*, rather than penalize, additional alignments.

3 Problem Formulation

In an *alignment problem* each input consists of a set of x variables L and a set of y variables R. For each input (L, R) there is a target alignment consisting of edges between individual x and individual y variables: $A \subseteq L \times R$. Note that an individual x or y variable may occur many times in such an alignment.

In a *structured* alignment problem, rather than taking on categorical or numerical values, each x variable encodes an element from a set of permissible structures $\mathcal{X}$, and each y value similarly represents a structure from $\mathcal{Y}$. For example, in our case the x variables are sentences in a text, and the y variables encode entries in a relational database. Thus, an input (L, R) represents a text-database pair for a particular NFL football game, and the target alignment A is the set of all sentence-entry pairs referring to the same fact or event.

Given a set of inputs and their true alignments, our goal is to learn a mapping from inputs to alignments which minimizes the errors in the predicted alignments.

4 Modeling Assumptions

Unfortunately, for a particular input (L, R) with $|L| = m$ and $|R| = n$, there are 2^{mn} possible alignments. For instance, in our scenario, where (L, R) is a text-database pair with around 50 sentences and 330 database entries, it is infeasible to search through this entire space. However, if some local decisions in the alignments are assumed to be independent of others, we can learn a model which makes optimal partial alignment decisions and then combines them into a full alignment. Next we describe several degrees of independence between local alignment decisions that might occur.

Full Independence In the case of full independence, the likelihood of any pair $(x, y) \in L \times R$ being aligned is independent of the alignment of any pair $(x', y') \in L \times R$ when *either* $x \neq x'$ *or* $y \neq y'$. Assuming this complete independence, we can train a local binary classifier using the $m \times n$ (x, y) pairs from each of our inputs.

Partial Independence A weaker, asymmetric independence might occur in one of two ways. Consider any two pairs $(x, y) \in L \times R$ and $(x', y') \in L \times R$. If the alignment decisions for these pairs are independent only if $x \neq x'$ then we say that the alignments are *left-side* partially independent. If, on the other hand, the alignment decisions are independent only if $y \neq y'$, then the alignments are *right-side* partially independent.

To illustrate these ideas, consider their application to our task of text-database alignment. Left-side partial independence would occur if the set of entries verbalized in a sentence is independent of those chosen for any *other* sentence. But at the same time, the entries to be verbalized in a *particular* sentence are chosen jointly and dependently – if, for example, a sentence mentioning a score is more likely now to mention a play which led to the score. We could thus train a model to choose the optimal aligned set of entries individually for each sentence.

5 Algorithm

To strike an appropriate balance between tractability and model richness, we base our text-database alignment method on a partial independence assumption. In particular, we assume that alignment decisions for each sentence are independent of other alignment decisions. At the same time, we allow dependencies within the set of database entries aligned to a sentence. Following the terminology introduced in Section 4, this model makes a left-side partial independence assumption. We believe that this is a natural choice for this task because a typical sentence conveys several facts that are semantically related in a meaningful way. When these facts are represented in a database, we can capture their inter-relations by examining the structural links of the corresponding database entries.

As a result of this independence assumption, we can train a model to make optimal alignment decisions separately for each sentence. In other words, each sentence is an independent instance that must be labeled with a subset of database entries. In essence, by treating entries as structured labels of sentences, we reformulate alignment as a structured multilabel classification task. The key advantage of our approach lies in its ability to consider arbitrary global features of each sentence and its proposed label set. However, since we cannot possibly examine every subset of labels, we need to intelligently prune the decision space. This pruning is driven by local alignments: for each sentence, all the database entries are ranked by their individual alignment likelihood. We then only consider sets of entries consistent with the ranking. That is, for every predicted label, the labels ranked above it must be predicted as well. Now that the number of considered candidates is linear in the number of entries, we can train a model to choose the one with optimal global characteristics. The latter stage of this approach is similar in spirit to global parse reranking [Collins and Koo, 2005]. However, our approach constrains the decision space through local ranking of labels rather than an n-best list produced by a generative model.

Below we formally describe the training and decoding procedures for text-database alignment.

5.1 Model Structure

The model takes as an input a set of sentences L with $|L| = m$ and a set of database entries R with $|R| = n$. Each sentence-entry pair $(x, y) \in L \times R$ is represented as a local feature vector $\Phi_{\mathbf{x},\mathbf{y}}(x, y)$. Each component of this vector is a binary feature. For instance, the i^{th} component of this vector may indicate whether the sentence and entry contain the same number:

$$(\Phi_{\mathbf{x},\mathbf{y}}(x, y))_i = \begin{cases} 1 & \text{if } x \text{ and } y \text{ contain the same number} \\ 0 & \text{otherwise} \end{cases}$$

In addition, our model captures dependencies among *multiple* entries aligned to the same sentence. We represent a sentence $x \in L$ paired with a set of candidate entries $Y \subseteq R$ with the global feature vector $\Phi_{\mathbf{x}}(x, Y)$. The components here are also binary features. For example, the i^{th} may indicate whether all the entries in Y match the same name in the sentence:

$$(\Phi_{\mathbf{x}}(x, Y))_i = \begin{cases} 1 & \forall y \in Y \text{ contain the same name} \\ 0 & \text{otherwise} \end{cases}$$

Our model predicts alignments in two basic steps.

Step One The goal of this step is to prune the decision space by inducing a ranking of database entries for each sentence $x \in L$. To do so, for each each entry $y \in R$, the model maps the feature vector $\Phi_{\mathbf{x},\mathbf{y}}(x, y)$ to a real number. This mapping can take the form of a linear model,

$$\Phi_{\mathbf{x},\mathbf{y}}(x, y) \cdot \alpha_1$$

where α_1 is the local parameter vector.

We then order the entries in R: $y_1, y_2, y_3, \ldots, y_n$ such that $\Phi_{\mathbf{x},\mathbf{y}}(x, y_i) \cdot \alpha_1 \geq \Phi_{\mathbf{x},\mathbf{y}}(x, y_{i+1}) \cdot \alpha_1, \forall i$.

Step Two Given a particular ranking of entries $y_1, y_2, \ldots, y_n$, for each $i \in 0, 1, ..., n$, we define the set of i initial elements $Y_i = \{y_1, \ldots, y_i\}$. The goal of this step is to choose a cut-off i^* and return Y_{i^*} as the aligned set of entries. To do so, our model computes a score for each $i \in 0, 1, \ldots, n$ based on the global feature vector $\Phi_{\mathbf{x}}(x, Y_i)$

$$\Phi_{\mathbf{x}}(x, Y_i) \cdot \alpha_2$$

where α_2 is the global parameter vector. The final output of the model is Y_{i^*}, for $i^* = \arg\max_i \Phi_{\mathbf{x}}(x, Y_i) \cdot \alpha_2$.

5.2 Training the Model

We are given a training corpus with text-database pairs (L, R) and alignments $A \subseteq L \times R$.

Training for Step One For the label ranking model $\Phi_{\mathbf{x},\mathbf{y}}(x, y) \cdot \alpha_1$, the scores returned by a classifier or ranking model may be used. Standard training techniques can be applied to learn this model. In fact, we employed an SVM classifier with a quadratic kernel as it yielded the best results for our task.

Training for Step Two The global model $\Phi_{\mathbf{x}}(x, Y_i) \cdot \alpha_2$ is trained using the label ranking returned by the local model. For each sentence $x \in L$ it considers all entry sets $Y_0, ..., Y_n$ formed by varying the cut-off point of the initial ranking. However, the true alignment may not be among these sets: the ranking may have incorrectly placed an unaligned entry above an aligned entry, in effect pruning away the true target. Therefore, to identify a new target alignment set Y_{i^*}, we need to identify the closest remaining entry set.

As our similarity metric between two sets A and B, we use the Hamming distance $H(A, B)$ (defined as the cardinality of the symmetric difference of A and B). For each input (L, R), and each sentence $x \in L$, we identify the cut-off in the ranking which produces the entry set closest in Hamming distance to the true label set T:

$$i^* = \arg\min_i H(Y_i, T)$$

This process of identifying a training target for the global model is illustrated in Table 1.

Once all the training targets in the corpus have been identified through this procedure, the global linear model is learned using a variant of the Perceptron algorithm [Collins and Duffy, 2002]. This algorithm encourages a setting of the parameter vector α_2 that will assign the highest score to the global feature vector associated with the optimal cut-off i^*.

$\alpha_2 \leftarrow 0$
`For each` (L, R) `and each sentence` $x \in L$:
  $i^{max} \leftarrow \arg\max_i \Phi_{\mathbf{x}}(x, Y_i) \cdot \alpha_2$
  `if` $i^{max} \neq i^*$`, set:`
    $\alpha_2 \leftarrow \alpha_2 + \Phi_{\mathbf{x}}(x, Y_{i^*})$
    $\alpha_2 \leftarrow \alpha_2 - \Phi_{\mathbf{x}}(x, Y_{i^{max}})$

		$\leftarrow$	$\Phi_{\mathbf{x}}(x, \{\})$	$H = 2$
-	y_1	$\leftarrow$	$\Phi_{\mathbf{x}}(x, \{y_1\})$	$H = 3$
+	y_2	$\leftarrow$	$\Phi_{\mathbf{x}}(x, \{y_1, y_2\})$	$H = 2$
+	y_3	$\leftarrow *$	$\Phi_{\mathbf{x}}(x, \{y_1, y_2, y_3\})$	$H = 1$
-	y_4	$\leftarrow$	$\Phi_{\mathbf{x}}(x, \{y_1, y_2, y_3, y_4\})$	$H = 2$

Table 1: To select the training target for the global model, we select the cut-off i^* which minimizes the Hamming distance H to the true label set. "+" and "-" indicate whether a label is in the true label set or not. In this example, $i^* = 3$, since the set $\{y_1, y_2, y_3\}$ has the minimum Hamming distance.

6 Features

In this section we describe local and global features used by our model.

Local Features

We assume that a high overlap in names and numbers increases the likelihood that a given sentence-entry pair is aligned. Thus, our local alignment is driven by anchor-based matching. A feature vector generated for each sentence-entry pair captures various statistics about name and number overlap. To further refine the matching process, these features also represent the unigrams and bigrams surrounding each matched name and number. In addition, the feature vector includes the type of database entry as certain entry types (i.e., *scoring_summary*) are more commonly verbalized in text. Close to 60,000 local features are generated. See Table 2 for a list of local feature templates.

Global Features

Our global model jointly considers a sentence with an entire set of candidate entries, seeking alignments with favorable structural properties. We select global features that express these properties, thereby allowing our model to correct local decisions. Two simple global features that we used are the number of aligned entries and the Hamming distance between aligned entries and the set predicted by a local model. We can group the remaining global features into three classes based on the type of dependency they capture:

- **Co-occurrence of entry types** A simple analysis of the aligned corpus reveals that certain entry types tend to be aligned together to the same sentence. For instance, entries from the *play-by-play* table are often verbalized together with entries from the *scoring-summary* table. To model this property, we introduce features for each commonly occurring set of entry types. These features parallel the label-pair co-occurrences modeled in Ghamrawi and McCallum [2005], but are not limited to pairs.
- **Local match overlap** A common error of local alignments is overmatching. A number that appears once in a sentence may be erroneously matched to multiple database entries during local alignment. For instance, consider the example in Figure 2: the number 4 in the sentence causes an overmatch with two database entries that contain this number. By examining the entire set

NAME_MATCH=(name category in entry)_LEX=(bigrams)
NUM_MATCH=(num category in entry)_LEX=(bigrams)
ENTRY_TYPE=(entry type)
COUNT_NUM_MATCHES=(count of num matches)
COUNT_NAME_MATCHES=(count of name matches)
%NUMS=(percent of nums in entry matched)
%NAMES=(percent of names in entry matched)

Table 2: Local feature templates. Text enclosed by brackets is replaced as appropriate. Also, various backoffs are used, such as using unigrams or no lexicalization at all in the first two features.

COUNT=(number of entries in set)
DIFF=(hamming distance to local prediction)
ENTRY_TYPES=(combination of entry types)
PLAYS_ALONE=(plays w/o corresponding scores)
SCORES_ALONE=(scores w/o corresponding plays)
PAIRS=(corresponding play/score pairs)
MATCHED_DRIVE=(drive with its plays or score)
UNMATCHED_DRIVE=(drive with other plays or scores)
MATCHED1_ENTRIES=(# entries matched by num)
MATCHED2_ENTRIES=(# entries matched by name)
SHARE1_ENTRIES=(# entries sharing nums)
SHARE2_ENTRIES=(# entries sharing name)
UNMATCHED1_ENTRIES=(# entries unmatched by num)
UNMATCHED2_ENTRIES=(# entries unmatched by name)
MATCHED_NUMS=(# of matched nums in sentence)
SHARED_NUMS=(# of shared nums in sentence)
MATCHED_NAMES=(# of matched names in sentence)
SHARED_NAMES=(# of shared names in sentence)

Table 3: Global feature templates.

of proposed entries simultaneously, we can separately count the entries that match *distinct* sentence anchors as well as those that *share* anchors. The latter number indicates the degree of overmatching and our model should learn to discourage such erroneous alignments.

- **Domain-specific semantic constraints** Besides these generic global features, we further refine the alignments using the database semantics defined by its schema. We can express a semantic relation between entries by analyzing their structural links in the database. For example, when summarizing the result of a *drive*, any play that occurred during the drive is likely to be mentioned as well. While the type co-occurrence feature described above would capture the correlation between these two entry *types*, it would fail to distinguish this case from semantically unrelated drives and plays. We engineered several domain-specific features of this variety.

Overall, 247 global features were generated. See Table 3 for a complete list of the global feature templates.

	Precision	Recall	F-measure
SVM baseline	84.33%	70.67%	76.90%
Multilabel Global	**87.26%**	**74.48%**	**80.31%**
Threshold Oracle	94.13%	85.76%	89.75%
Multilabel Regression	77.65%	64.08%	70.21%
Graph Matching	73.36%	64.47%	68.63%

Table 4: 10-fold cross validation results for the baseline, our global multilabel model, an oracle, and two other approaches.

7 Evaluation Setup

Corpus For our evaluation, we used the NFL football corpus previously used by Barzilay and Lapata [2005]. This corpus contains text-database pairs for 466 football games played over the 2003 and 2004 seasons. The database contains 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. This information is organized into several table types including *aggregate statistics*, *scoring_summary*, *team_comparison*, *drivechart*, and *playbyplay*. Each game is typically represented by 330 database entries. The accompanying texts are written by Associated Press journalists and typically contain 50 sentences.

Annotation In order to train and evaluate our model, we had a human annotator explicitly mark the linked sentence-entry pairs for a set of 78 games. We also conducted a human interannotator agreement study on a smaller set of 10 games. We compute a Kappa statistic over the chance of agreement for both positive and negative alignment decisions. We found high agreement between annotators, yielding a Kappa score of 0.73.

On average, 24 sentence-entry pairs (out of 330×50) are aligned per game. These alignments include about 6% of all database entries and about 28% of all sentences. Most aligned sentences are aligned to more than one database entry.

Training and Testing Regime From 78 manually annotated texts, we obtain about 1,250,000 entry-sentence pairs, from which 1,900 represent positive alignments. Given the relatively low number of positive examples, we opted to use 10-way cross validation for training and testing our algorithm. We compute precision and recall using aligned pairs as positive examples and unaligned pairs as negative examples.

8 Results

The results of the evaluation are shown in Table 4. We first compare our global model to a local classifier. This baseline model makes independent alignment decisions for each sentence-entry pair and is implemented using an SVM classifier with a quadratic kernel. The local features used are those described in Section 6. Our method achieves a 3.4% absolute reduction in error which corresponds to a 15% relative reduction in error. The global model predicts a different alignment for 239 out of 3,732 sentences in the corpus. Of these changes, 170 are improvements over the local model, while in

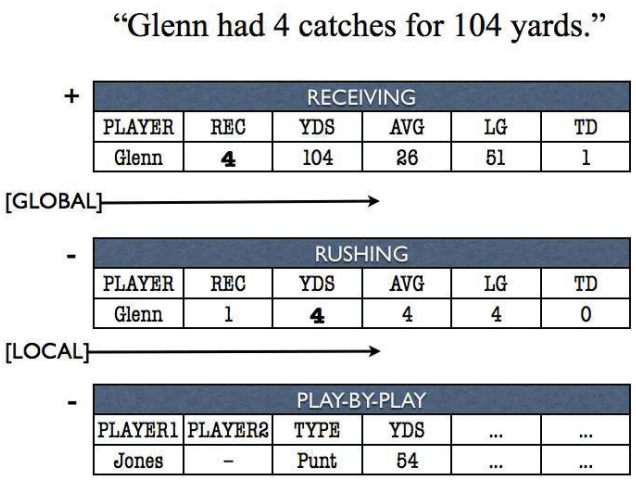

Figure 2: The top three ranked entries for the given sentence. The local model erroneously aligns the top two due to the matched "4", but the global model corrects this overmatching.

the other 69 cases the accuracy goes down. By a sign test, this difference is statistically significant at the level of $p < 0.01$. Figure 2 shows an example where global features correct the erroneous decisions of the local alignment.

We also present the performance obtained when an oracle chooses the optimal cut-off of the possibly flawed local label ranking. The performance of the oracle compared to our global model indicates that we can further improve alignment by either refining our search of the global space or by considering more powerful global features.

We also compare our model against a standard multilabel classifier [Elisseeff and Weston, 2001]. Like our model, this method first ranks the labels by their local scores, but then applies regression on these scores to determine an optimal threshold. This approach fails to outperform the SVM baseline.

Finally, we compare against the global alignment model of Taskar *et al.* [2005]. This model was developed for the task of word alignment, and attempts to maximize the local scores of the aligned word pairs subject to a fertility constraint. The best results we have obtained for this model are lower than the quadratic kernel SVM baseline. We used fertility constraints that allowed database entries to match up to three times and sentences to match up to five times. While we attempted to find the best settings for these and other parameter values, we cannot guarantee that the values selected are fully optimal, as our application is quite different from the model's original setting.

9 Conclusions

This paper introduces a novel algorithm for aligning database entries to the sentences that verbalize their content. We cast text-database alignment as structured multilabel classification. In contrast to existing multilabel classifiers, our method operates over arbitrary global features of inputs and proposed labels. Our empirical results show that this model yields superior performance without sacrificing tractability.

Acknowledgments

The authors acknowledge the support of the National Science Foundation (Barzilay; CAREER grant IIS-0448168 and grant IIS-0415865). Thanks to Eli Barzilay, Michael Collins, Pawan Deshpande, Yoong Keok Lee, Igor Malioutov, 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 NSF.

References

[Barzilay and Lapata, 2005] Regina Barzilay and Mirella Lapata. Collective content selection for concept-to-text generation. In *Proceedings of HLT/EMNLP*, pages 331–338, Vancouver, 2005.

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

[Collins and Duffy, 2002] Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the ACL*, 2002.

[Collins and Koo, 2005] Michael Collins and Terry Koo. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69, 2005.

[Crammer and Singer, 2002] Koby Crammer and Yoram Singer. A new family of online algorithms for category ranking. In *Proceedings of SIGIR*, pages 151–158, 2002.

[Elisseeff and Weston, 2001] Andre Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Proceeding of NIPS*, pages 681–687, 2001.

[Ghamrawi and McCallum, 2005] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *Proceedings of CIKM*, pages 195–200, 2005.

[Lacoste-Julien *et al.*, 2006] Simon Lacoste-Julien, Ben Taskar, Dan Klein, and Michael Jordan. Word alignment via quadratic assignment. In *Proceedings of HLT/NAACL*, 2006.

[McDonald *et al.*, 2005] Ryan McDonald, Koby Crammer, and Fernando Pereira. Flexible text segmentation with structured multilabel classification. In *Proceedings of HLT/EMNLP*, pages 987–994, 2005.

[Schapire and Singer, 1999] Robert E. Schapire and Yoram Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[Taskar *et al.*, 2005] Ben Taskar, Simon Lacoste-Julien, and Dan Klein. A discriminative matching. approach to word alignment. In *Proceedings of HLT/EMNLP*, pages 73–80, 2005.