Learning Semantic Correspondences with Less Supervision

Percy Liang
UC Berkeley
pliang@cs.berkeley.edu

Michael I. Jordan
UC Berkeley
jordan@cs.berkeley.edu

Dan Klein
UC Berkeley
klein@cs.berkeley.edu

Abstract

A central problem in grounded language acquisition is learning the correspondences between a rich world state and a stream of text which references that world state. To deal with the high degree of ambiguity present in this setting, we present a generative model that simultaneously segments the text into utterances and maps each utterance to a meaning representation grounded in the world state. We show that our model generalizes across three domains of increasing difficulty—Robocup sportscasting, weather forecasts (a new domain), and NFL recaps.

1 Introduction

Recent work in learning semantics has focused on mapping sentences to meaning representations (e.g., some logical form) given aligned sentence/meaning pairs as training data (Ge and Mooney, 2005; Zettlemoyer and Collins, 2005; Zettlemoyer and Collins, 2007; Lu et al., 2008). However, this degree of supervision is unrealistic for modeling human language acquisition and can be costly to obtain for building large-scale, broad-coverage language understanding systems.

A more flexible direction is grounded language acquisition: learning the meaning of sentences in the context of an observed world state. The grounded approach has gained interest in various disciplines (Siskind, 1996; Yu and Ballard, 2004; Feldman and Narayanan, 2004; Gorniak and Roy, 2007). Some recent work in the NLP community has also moved in this direction by relaxing the amount of supervision to the setting where each sentence is paired with a small set of candidate meanings (Kate and Mooney, 2007; Chen and Mooney, 2008).

The goal of this paper is to reduce the amount of supervision even further. We assume that we are given a *world state* represented by a set of *records* along with a *text*, an unsegmented sequence of words. For example, in the weather forecast domain (Section 2.2), the text is the weather report, and the records provide a structured representation of the temperature, sky conditions, etc.

In this less restricted data setting, we must resolve multiple ambiguities: (1) the segmentation of the text into *utterances*; (2) the identification of relevant *facts*, i.e., the choice of records and aspects of those records; and (3) the alignment of utterances to facts (facts are the meaning representations of the utterances). Furthermore, in some of our examples, much of the world state is not referenced at all in the text, and, conversely, the text references things which are not represented in our world state. This increased amount of ambiguity and noise presents serious challenges for learning. To cope with these challenges, we propose a probabilistic generative model that treats text segmentation, fact identification, and alignment in a single unified framework. The parameters of this hierarchical hidden semi-Markov model can be estimated efficiently using EM.

We tested our model on the task of aligning text to records in three different domains. The first domain is Robocup sportscasting (Chen and Mooney, 2008). Their best approach (KRISPER) obtains 67% F_1; our method achieves 76.5%. This domain is simplified in that the segmentation is known. The second domain is weather forecasts, for which we created a new dataset. Here, the full complexity of joint segmentation and alignment arises. Nonetheless, we were able to obtain reasonable results on this task. The third domain we considered is NFL recaps (Barzilay and Lapata, 2005; Snyder and Barzilay, 2007). The language used in this domain is richer by orders of magnitude, and much of it does *not* reference the world state. Nonetheless, taking the first unsupervised approach to this problem, we were able to make substantial progress: We achieve an F_1 of 53.2%, which closes over half of the gap between a heuristic baseline (26%) and supervised systems (68%–80%).

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 91–99,
Suntec, Singapore, 2-7 August 2009.

Dataset	# scenarios	$\lvert\mathbf{w}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathbf{s}\rvert$	$\lvert\mathcal{A}\rvert$
Robocup	1919	5.7	9	2.4	0.8
Weather	22146	28.7	12	36.0	5.8
NFL	78	969.0	44	329.0	24.3

Table 1: Statistics for the three datasets. We report average values across all scenarios in the dataset: $\lvert\mathbf{w}\rvert$ is the number of words in the text, $\lvert\mathcal{T}\rvert$ is the number of record types, $\lvert\mathbf{s}\rvert$ is the number of records, and $\lvert\mathcal{A}\rvert$ is the number of gold alignments.

2 Domains and Datasets

Our goal is to learn the correspondence between a text $\mathbf{w}$ and the world state $\mathbf{s}$ it describes. We use the term *scenario* to refer to such a $(\mathbf{w}, \mathbf{s})$ pair.

The *text* is simply a sequence of words $\mathbf{w} = (w_1, \ldots, w_{|\mathbf{w}|})$. We represent the world state $\mathbf{s}$ as a set of *records*, where each *record* $r \in \mathbf{s}$ is described by a *record type* $r.t \in \mathcal{T}$ and a tuple of *field values* $r.\mathbf{v} = (r.v_1, \ldots, r.v_m)$.[1] For example, temperature is a record type in the weather domain, and it has four fields: time, min, mean, and max.

The record type $r.t \in \mathcal{T}$ specifies the *field type* $r.t_f \in \{\text{INT}, \text{STR}, \text{CAT}\}$ of each field value $r.v_f$, $f = 1, \ldots, m$. There are three possible field types—integer (INT), string (STR), and categorical (CAT)—which are assumed to be known and fixed. Integer fields represent numeric properties of the world such as temperature, string fields represent surface-level identifiers such as names of people, and categorical fields represent discrete concepts such as score types in football (touchdown, field goal, and safety). The field type determines the way we expect the field value to be rendered in words: integer fields can be numerically perturbed, string fields can be spliced, and categorical fields are represented by open-ended word distributions, which are to be learned. See Section 3.3 for details.

2.1 Robocup Sportscasting

In this domain, a Robocup simulator generates the state of a soccer game, which is represented by a set of event records. For example, the record pass(arg1=`pink1`,arg2=`pink5`) denotes a passing event; this type of record has two fields: arg1 (the actor) and arg2 (the recipient). As the game is progressing, humans interject commentaries about notable events in the game, e.g., *pink1 passes back to pink5 near the middle of the field*. All of the fields in this domain are categorical, which means there is no a priori association between the field value `pink1` and the word *pink1*. This degree of flexibility is desirable because *pink1* is sometimes referred to as *pink goalie*, a mapping which does not arise from string operations but must instead be learned.

We used the dataset created by Chen and Mooney (2008), which contains 1919 scenarios from the 2001–2004 Robocup finals. Each scenario consists of a single sentence representing a fragment of a commentary on the game, paired with a set of candidate records. In the annotation, each sentence corresponds to at most one record (possibly one not in the candidate set, in which case we automatically get that sentence wrong). See Figure 1(a) for an example and Table 1 for summary statistics on the dataset.

2.2 Weather Forecasts

In this domain, the world state contains detailed information about a local weather forecast and the text is a short forecast report (see Figure 1(b) for an example). To create the dataset, we collected local weather forecasts for 3,753 cities in the US (those with population at least 10,000) over three days (February 7–9, 2009) from `www.weather.gov`. For each city and date, we created two scenarios, one for the day forecast and one for the night forecast. The forecasts consist of hour-by-hour measurements of temperature, wind speed, sky cover, chance of rain, etc., which represent the underlying world state.

This world state is summarized by records which aggregate measurements over selected time intervals. For example, one of the records states the minimum, average, and maximum temperature from 5pm to 6am. This aggregation process produced 22,146 scenarios, each containing $|\mathbf{s}| = 36$ multi-field records. There are 12 record types, each consisting of only integer and categorical fields.

To annotate the data, we split the text by punctuation into *lines* and labeled each line with the records to which the line refers. These lines are used only for evaluation and are not part of the model (see Section 5.1 for further discussion).

The weather domain is more complex than the Robocup domain in several ways: The text $\mathbf{w}$ is longer, there are more candidate records, and most notably, $\mathbf{w}$ references multiple records (5.8 on av-

[1] To simplify notation, we assume that each record has m fields, though in practice, m depends on the record type $r.t$.

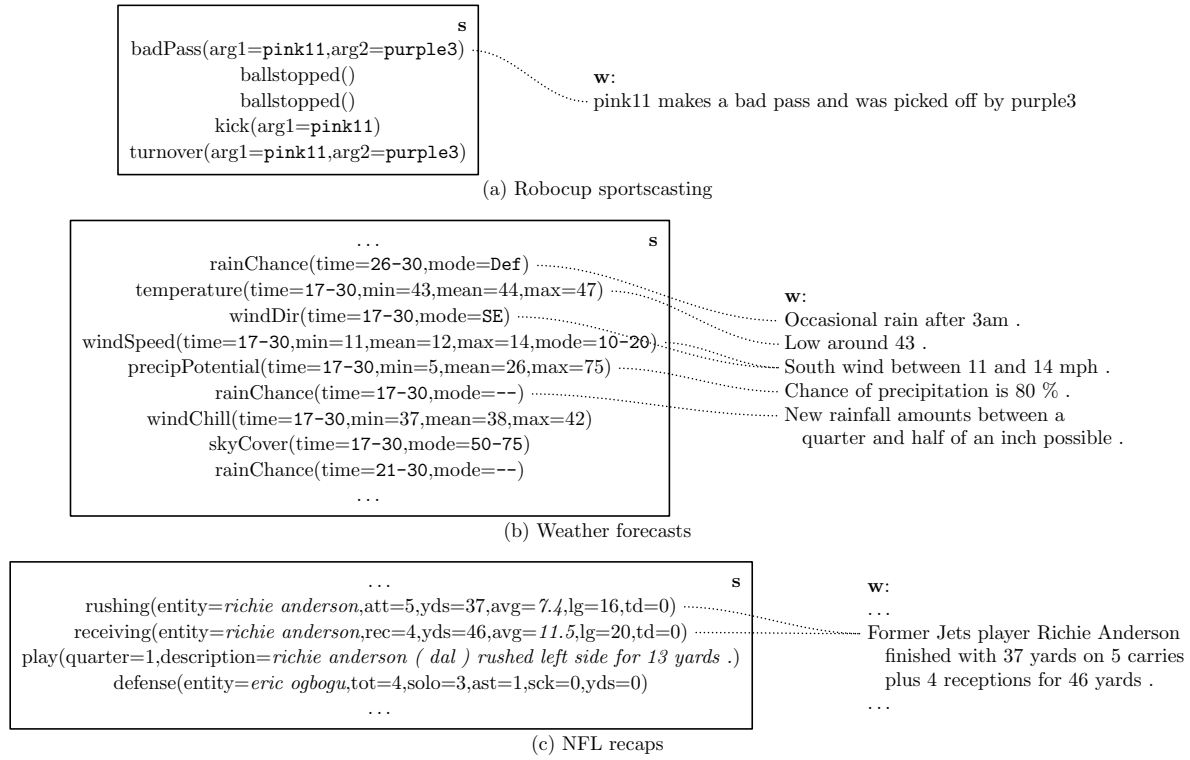


Figure 1: An example of a scenario for each of the three domains. Each scenario consists of a candidate set of records $\mathbf{s}$ and a text $\mathbf{w}$. Each record is specified by a record type (e.g., badPass) and a set of field values. Integer values are in Roman, string values are in *italics*, and categorical values are in `typewriter`. The gold alignments are shown.

erage), so the segmentation of $\mathbf{w}$ is unknown. See Table 1 for a comparison of the two datasets.

2.3 NFL Recaps

In this domain, each scenario represents a single NFL football game (see Figure 1(c) for an example). The world state (the things that happened during the game) is represented by database tables, e.g., scoring summary, team comparison, drive chart, play-by-play, etc. Each record is a database entry, for instance, the receiving statistics for a certain player. The text is the recap of the game—an article summarizing the game highlights. The dataset we used was collected by Barzilay and Lapata (2005). The data includes 466 games during the 2003–2004 NFL season. 78 of these games were annotated by Snyder and Barzilay (2007), who aligned each sentence to a set of records.

This domain is by far the most complicated of the three. Many records corresponding to inconsequential game statistics are not mentioned. Conversely, the text contains many general remarks (e.g., *it was just that type of game*) which are not present in any of the records. Furthermore, the complexity of the language used in the recap is far greater than what we can represent using our simple model. Fortunately, most of the fields are integer fields or string fields (generally names or brief descriptions), which provide important anchor points for learning the correspondences. Nonetheless, the same names and numbers occur in multiple records, so there is still uncertainty about which record is referenced by a given sentence.

3 Generative Model

To learn the correspondence between a text $\mathbf{w}$ and a world state $\mathbf{s}$, we propose a generative model $p(\mathbf{w} \mid \mathbf{s})$ with latent variables specifying this correspondence.

Our model combines segmentation with alignment. The segmentation aspect of our model is similar to that of Grenager et al. (2005) and Eisenstein and Barzilay (2008), but in those two models, the segments are clustered into topics rather than grounded to a world state. The alignment aspect of our model is similar to the HMM model for word alignment (Ney and Vogel, 1996). DeNero et al. (2008) perform joint segmentation and word alignment for machine translation, but the nature of that task is different from ours.

The model is defined by a generative process,

which proceeds in three stages (Figure 2 shows the corresponding graphical model):

1. Record choice: choose a sequence of records $\mathbf{r} = (r_1, \ldots, r_{|\mathbf{r}|})$ to describe, where each $r_i \in \mathbf{s}$.
2. Field choice: for each chosen record r_i, select a sequence of fields $\mathbf{f}_i = (f_{i1}, \ldots, f_{i|\mathbf{f}_i|})$, where each $f_{ij} \in \{1, \ldots, m\}$.
3. Word choice: for each chosen field f_{ij}, choose a number $c_{ij} > 0$ and generate a sequence of c_{ij} words.

The observed text $\mathbf{w}$ is the terminal yield formed by concatenating the sequences of words of all fields generated; note that the segmentation of $\mathbf{w}$ provided by $\mathbf{c} = \{c_{ij}\}$ is latent. Think of the words spanned by a record as constituting an utterance with a meaning representation given by the record and subset of fields chosen.

Formally, our probabilistic model places a distribution over $(\mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{w})$ and factorizes according to the three stages as follows:

$$p(\mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{w} \mid \mathbf{s}) = p(\mathbf{r} \mid \mathbf{s}) p(\mathbf{f} \mid \mathbf{r}) p(\mathbf{c}, \mathbf{w} \mid \mathbf{r}, \mathbf{f}, \mathbf{s})$$

The following three sections describe each of these stages in more detail.

3.1 Record Choice Model

The record choice model specifies a distribution over an ordered sequence of records $\mathbf{r} = (r_1, \ldots, r_{|\mathbf{r}|})$, where each record $r_i \in \mathbf{s}$. This model is intended to capture two types of regularities in the discourse structure of language. The first is *salience*, that is, some record types are simply more prominent than others. For example, in the NFL domain, 70% of scoring records are mentioned whereas only 1% of punting records are mentioned. The second is the idea of local *coherence*, that is, the order in which one mentions records tend to follow certain patterns. For example, in the weather domain, the sky conditions are generally mentioned first, followed by temperature, and then wind speed.

To capture these two phenomena, we define a Markov model on the record types (and given the record type, a record is chosen uniformly from the set of records with that type):

$$p(\mathbf{r} \mid \mathbf{s}) = \prod_{i=1}^{|\mathbf{r}|} p(r_i.t \mid r_{i-1}.t) \frac{1}{|\mathbf{s}(r_i.t)|}, \qquad (1)$$

where $\mathbf{s}(t) \stackrel{\text{def}}{=} \{r \in \mathbf{s} : r.t = t\}$ and $r_0.t$ is a dedicated START record type.[2] We also model the transition of the final record type to a designated STOP record type in order to capture regularities about the types of records which are described last. More sophisticated models of coherence could also be employed here (Barzilay and Lapata, 2008).

We assume that $\mathbf{s}$ includes a special *null record* whose type is NULL, responsible for generating parts of our text which do not refer to any real records.

3.2 Field Choice Model

Each record type $t \in \mathcal{T}$ has a separate field choice model, which specifies a distribution over a sequence of fields. We want to capture salience and coherence at the field level like we did at the record level. For instance, in the weather domain, the minimum and maximum fields of a temperature record are mentioned whereas the average is not. In the Robocup domain, the actor typically precedes the recipient in passing event records.

Formally, we have a Markov model over the fields:[3]

$$p(\mathbf{f} \mid \mathbf{r}) = \prod_{i=1}^{|\mathbf{r}|} \prod_{j=1}^{|\mathbf{f}_j|} p(f_{ij} \mid f_{i(j-1)}). \qquad (2)$$

Each record type has a dedicated *null field* with its own multinomial distribution over words, intended to model words which refer to that record type in general (e.g., the word *passes* for passing records). We also model transitions into the first field and transitions out of the final field with special START and STOP fields. This Markov structure allows us to capture a few elements of rudimentary syntax.

3.3 Word Choice Model

We arrive at the final component of our model, which governs how the information about a particular field of a record is rendered into words. For each field f_{ij}, we generate the number of words c_{ij} from a uniform distribution over $\{1, 2, \ldots, C_{\text{max}}\}$, where C_{max} is set larger than the length of the longest text we expect to see. Conditioned on

[2] We constrain our inference to only consider record types t that occur in $\mathbf{s}$, i.e., $\mathbf{s}(t) \neq \emptyset$.

[3] During inference, we prohibit consecutive fields from repeating.

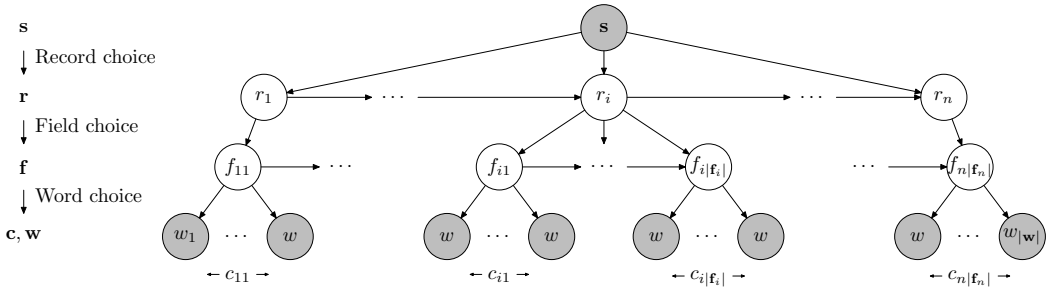


Figure 2: Graphical model representing the generative model. First, records are chosen and ordered from the set $\mathbf{s}$. Then fields are chosen for each record. Finally, words are chosen for each field. The world state $\mathbf{s}$ and the words $\mathbf{w}$ are observed, while $(\mathbf{r}, \mathbf{f}, \mathbf{c})$ are latent variables to be inferred (note that the number of latent variables itself is unknown).

the fields $\mathbf{f}$, the words $\mathbf{w}$ are generated independently:[4]

$$p(\mathbf{w} \mid \mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{s}) = \prod_{k=1}^{|\mathbf{w}|} p_{\mathrm{W}}(w_k \mid r(k).t_{f(k)}, r(k).v_{f(k)}),$$

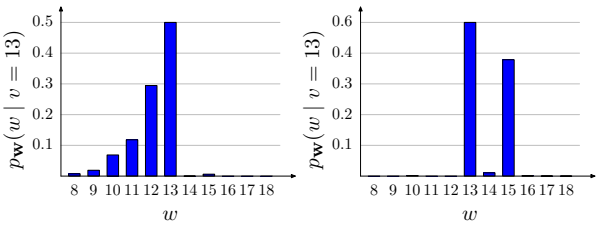
where $r(k)$ and $f(k)$ are the record and field responsible for generating word w_k, as determined by the segmentation $\mathbf{c}$. The word choice model $p_{\mathrm{W}}(w \mid t, v)$ specifies a distribution over words given the field type t and field value v. This distribution is a mixture of a global backoff distribution over words and a field-specific distribution which depends on the field type t.

Although we designed our word choice model to be relatively general, it is undoubtedly influenced by the three domains. However, we can readily extend or replace it with an alternative if desired; this modularity is one principal benefit of probabilistic modeling.

Integer Fields ($t = \textsc{int}$) For integer fields, we want to capture the intuition that a numeric quantity v is rendered in the text as a word which is possibly some other numerical value w due to stylistic factors. Sometimes the exact value v is used (e.g., in reporting football statistics). Other times, it might be customary to round v (e.g., wind speeds are typically rounded to a multiple of 5). In other cases, there might just be some unexplained error, where w deviates from v by some noise $\epsilon_+ = w - v > 0$ or $\epsilon_- = v - w > 0$. We model ϵ_+ and ϵ_- as geometric distributions.[5] In summary, we allow six possible ways of generating the word w given v:

$$v \quad \lceil v \rceil_5 \quad \lfloor v \rfloor_5 \quad \text{round}_5(v) \quad v - \epsilon_- \quad v + \epsilon_+$$

Separate probabilities for choosing among these possibilities are learned for each field type (see Figure 3 for an example).



(a) temperature.min (b) windSpeed.min

Figure 3: Two integer field types in the weather domain for which we learn different distributions over the ways in which a value v might appear in the text as a word w. Suppose the record field value is $v = 13$. Both distributions are centered around v, as is to be expected, but the two distributions have different shapes: For temperature.min, almost all the mass is to the left, suggesting that forecasters tend to report conservative lower bounds. For the wind speed, the mass is concentrated on 13 and 15, suggesting that forecasters frequently round wind speeds to multiples of 5.

String Fields ($t = \textsc{str}$) Strings fields are intended to represent values which we expect to be realized in the text via a simple surface-level transformation. For example, a name field with value $v =$ *Moe Williams* is sometimes referenced in the text by just *Williams*. We used a simple generic model of rendering string fields: Let w be a word chosen uniformly from those in v.

Categorical Fields ($t = \textsc{cat}$) Unlike string fields, categorical fields are not tied down to any lexical representation; in fact, the identities of the categorical field values are irrelevant. For each categorical field f and possible value v, we have a

[4] While a more sophisticated model of words would be useful if we intended to use this model for natural language generation, the false independence assumptions present here matter less for the task of learning the semantic correspondences because we always condition on $\mathbf{w}$.

[5] Specifically, $p(\epsilon_+; \alpha_+) = (1 - \alpha_+)^{\epsilon_+ - 1}\alpha_+$, where α_+ is a field-specific parameter; $p(\epsilon_-; \alpha_-)$ is defined analogously.

v	$p_{\mathrm{W}}(w \mid t, v)$
0-25	, clear mostly sunny
25-50	partly , cloudy increasing
50-75	mostly cloudy , partly
75-100	of inch an possible new a rainfall

Table 2: Highest probability words for the categorical field skyCover.mode in the weather domain. It is interesting to note that skyCover=75-100 is so highly correlated with rain that the model learns to connect an overcast sky in the world to the indication of rain in the text.

separate multinomial distribution over words from which w is drawn. An example of a categorical field is skyCover.mode in the weather domain, which has four values: 0-25, 25-50, 50-75, and 75-100. Table 2 shows the top words for each of these field values learned by our model.

4 Learning and Inference

Our learning and inference methodology is a fairly conventional application of Expectation Maximization (EM) and dynamic programming. The input is a set of scenarios $\mathcal{D}$, each of which is a text $\mathbf{w}$ paired with a world state $\mathbf{s}$. We maximize the marginal likelihood of our data, summing out the latent variables $(\mathbf{r}, \mathbf{f}, \mathbf{c})$:

$$\max_{\theta} \prod_{(\mathbf{w},\mathbf{s})\in\mathcal{D}} \sum_{\mathbf{r},\mathbf{f},\mathbf{c}} p(\mathbf{r}, \mathbf{f}, \mathbf{c}, \mathbf{w} \mid \mathbf{s}; \theta), \qquad (3)$$

where θ are the parameters of the model (all the multinomial probabilities). We use the EM algorithm to maximize (3), which alternates between the E-step and the M-step. In the E-step, we compute expected counts according to the posterior $p(\mathbf{r}, \mathbf{f}, \mathbf{c} \mid \mathbf{w}, \mathbf{s}; \theta)$. In the M-step, we optimize the parameters θ by normalizing the expected counts computed in the E-step. In our experiments, we initialized EM with a uniform distribution for each multinomial and applied add-0.1 smoothing to each multinomial in the M-step.

As with most complex discrete models, the bulk of the work is in computing expected counts under $p(\mathbf{r}, \mathbf{f}, \mathbf{c} \mid \mathbf{w}, \mathbf{s}; \theta)$. Formally, our model is a hierarchical hidden semi-Markov model conditioned on $\mathbf{s}$. Inference in the E-step can be done using a dynamic program similar to the inside-outside algorithm.

5 Experiments

Two important aspects of our model are the segmentation of the text and the modeling of the coherence structure at both the record and field levels. To quantify the benefits of incorporating these two aspects, we compare our full model with two simpler variants.

- Model 1 (no model of segmentation or coherence): Each record is chosen independently; each record generates one field, and each field generates one word. This model is similar in spirit to IBM model 1 (Brown et al., 1993).
- Model 2 (models segmentation but not coherence): Records and fields are still generated independently, but each field can now generate multiple words.
- Model 3 (our full model of segmentation and coherence): Records and fields are generated according to the Markov chains described in Section 3.

5.1 Evaluation

In the annotated data, each text $\mathbf{w}$ has been divided into a set of lines. These lines correspond to clauses in the weather domain and sentences in the Robocup and NFL domains. Each line is annotated with a (possibly empty) set of records. Let $\mathcal{A}$ be the gold set of these line-record alignment pairs.

To evaluate a learned model, we compute the Viterbi segmentation and alignment ($\mathrm{argmax}_{\mathbf{r},\mathbf{f},\mathbf{c}}\, p(\mathbf{r}, \mathbf{f}, \mathbf{c} \mid \mathbf{w}, \mathbf{s})$). We produce a predicted set of line-record pairs $\mathcal{A}'$ by aligning a line to a record r_i if the span of (the utterance corresponding to) r_i overlaps the line. The reason we evaluate indirectly using lines rather than using utterances is that it is difficult to annotate the segmentation of text into utterances in a simple and consistent manner.

We compute standard precision, recall, and F_1 of $\mathcal{A}'$ with respect to $\mathcal{A}$. Unless otherwise specified, performance is reported on all scenarios, which were also used for training. However, we did not tune any hyperparameters, but rather used generic values which worked well enough across all three domains.

5.2 Robocup Sportscasting

We ran 10 iterations of EM on Models 1–3. Table 3 shows that performance improves with increased model sophistication. We also compare

Method	Precision	Recall	F_1
Model 1	**78.6**	61.9	69.3
Model 2	74.1	**84.1**	78.8
Model 3	77.3	84.0	**80.5**

Table 3: Alignment results on the Robocup sportscasting dataset.

Method	F_1
Random baseline	48.0
Chen and Mooney (2008)	67.0
Model 3	**75.7**

Table 4: F_1 scores based on the 4-fold cross-validation scheme in Chen and Mooney (2008).

Method	Precision	Recall	F_1
Model 1	49.9	75.1	60.0
Model 2	67.3	70.4	68.8
Model 3	**76.3**	**73.8**	**75.0**

Table 5: Alignment results on the weather forecast dataset.

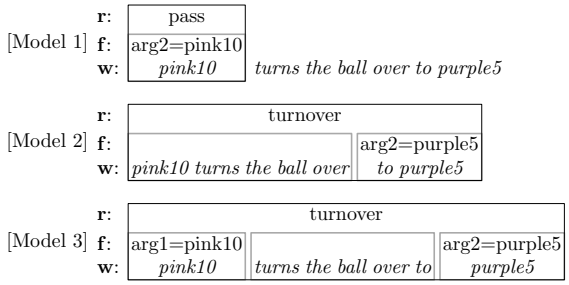


Figure 4: An example of predictions made by each of the three models on the Robocup dataset.

our model to the results of Chen and Mooney (2008) in Table 4.

Figure 4 provides a closer look at the predictions made by each of our three models for a particular example. Model 1 easily mistakes *pink10* for the recipient of a pass record because decisions are made independently for each word. Model 2 chooses the correct record, but having no model of the field structure inside a record, it proposes an incorrect field segmentation (although our evaluation is insensitive to this). Equipped with the ability to prefer a coherent field sequence, Model 3 fixes these errors.

Many of the remaining errors are due to the garbage collection phenomenon familiar from word alignment models (Moore, 2004; Liang et al., 2006). For example, the ballstopped record occurs frequently but is never mentioned in the text. At the same time, there is a correlation between ballstopped and utterances such as *pink2 holds onto the ball*, which are not aligned to any record in the annotation. As a result, our model incorrectly chooses to align the two.

5.3 Weather Forecasts

For the weather domain, staged training was necessary to get good results. For Model 1, we ran 15 iterations of EM. For Model 2, we ran 5 iterations of EM on Model 1, followed by 10 iterations on Model 2. For Model 3, we ran 5 iterations of Model 1, 5 iterations of a simplified variant of Model 3 where records were chosen independently, and finally, 5 iterations of Model 3. When going from one model to another, we used the final posterior distributions of the former to initialize the parameters of the latter.[6] We also prohibited utterances in Models 2 and 3 from crossing punctuation during inference.

Table 5 shows that performance improves substantially in the more sophisticated models, the gains being greater than in the Robocup domain. Figure 5 shows the predictions of the three models on an example. Model 1 is only able to form isolated (but not completely inaccurate) associations. By modeling segmentation, Model 2 accounts for the intermediate words, but errors are still made due to the lack of Markov structure. Model 3 remedies this. However, unexpected structures are sometimes learned. For example, the temperature.time=`6-21` field indicates daytime, which happens to be perfectly correlated with the word *high*, although *high* intuitively should be associated with the temperature.max field. In these cases of high correlation (Table 2 provides another example), it is very difficult to recover the proper alignment without additional supervision.

5.4 NFL Recaps

In order to scale up our models to the NFL domain, we first pruned for each sentence the records which have either no numerical values (e.g., 23, 23-10, 2/4) nor name-like words (e.g., those that appear only capitalized in the text) in common. This eliminated all but 1.5% of the record candidates per sentence, while maintaining an ora-

[6] It is interesting to note that this type of staged training is evocative of language acquisition in children: lexical associations are formed (Model 1) before higher-level discourse structure is learned (Model 3).

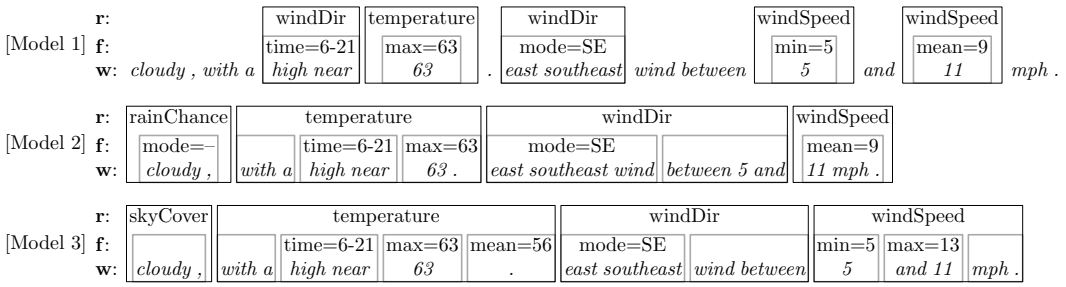


Figure 5: An example of predictions made by each of the three models on the weather dataset.

cle alignment F_1 score of 88.7. Guessing a single random record for each sentence yields an F_1 of 12.0. A reasonable heuristic which uses weighted number- and string-matching achieves 26.7.

Due to the much greater complexity of this domain, Model 2 was easily misled as it tried without success to find a coherent segmentation of the fields. We therefore created a variant, Model 2', where we constrained each field to generate exactly one word. To train Model 2', we ran 5 iterations of EM where each sentence is assumed to have exactly one record, followed by 5 iterations where the constraint was relaxed to also allow record boundaries at punctuation and the word *and*. We did not experiment with Model 3 since the discourse structure on records in this domain is not at all governed by a simple Markov model on record types—indeed, most regions do not refer to any records at all. We also fixed the backoff probability to 0.1 instead of learning it and enforced zero numerical deviation on integer field values.

Model 2' achieved an F_1 of 39.9, an improvement over Model 1, which attained 32.8. Inspection of the errors revealed the following problem: The alignment task requires us to sometimes align a sentence to multiple redundant records (e.g., play and score) referenced by the same part of the text. However, our model generates each part of text from only one record, and thus it can only allow an alignment to one record.[7] To cope with this incompatibility between the data and our notion of semantics, we used the following solution: We divided the records into three groups by type: play, score, and other. Each group has a copy of the model, but we enforce that they share the same segmentation. We also introduce a potential that couples the presence or absence of records across groups on the same segment to capture regular co-occurrences between redundant records.

[7]The model can align a sentence to multiple records provided that the records are referenced by non-overlapping parts of the text.

Method	Precision	Recall	F_1
Random (with pruning)	13.1	11.0	12.0
Baseline	29.2	24.6	26.7
Model 1	25.2	46.9	32.8
Model 2'	43.4	37.0	39.9
Model 2' (with groups)	46.5	62.1	53.2
Graph matching (sup.)	73.4	64.5	68.6
Multilabel global (sup.)	87.3	74.5	80.3

Table 6: Alignment results on the NFL dataset. Graph matching and multilabel are supervised results reported in Snyder and Barzilay (2007).[9]

Table 6 shows our results. With groups, we achieve an F_1 of 53.2. Though we still trail supervised techniques, which attain numbers in the 68–80 range, we have made substantial progress over our baseline using an unsupervised method. Furthermore, our model provides a more detailed analysis of the correspondence between the world state and text, rather than just producing a single alignment decision. Most of the remaining errors made by our model are due to a lack of calibration. Sometimes, our false positives are close calls where a sentence indirectly references a record, and our model predicts the alignment whereas the annotation standard does not. We believe that further progress is possible with a richer model.

6 Conclusion

We have presented a generative model of correspondences between a world state and an unsegmented stream of text. By having a joint model of salience, coherence, and segmentation, as well as a detailed rendering of the values in the world state into words in the text, we are able to cope with the increased ambiguity that arises in this new data setting, successfully pushing the limits of unsupervision.

References

R. Barzilay and M. Lapata. 2005. Collective content selection for concept-to-text generation. In *Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 331–338, Vancouver, B.C.

R. Barzilay and M. Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34:1–34.

P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

D. L. Chen and R. J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *International Conference on Machine Learning (ICML)*, pages 128–135. Omnipress.

J. DeNero, A. Bouchard-Côté, and D. Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 314–323, Honolulu, HI.

J. Eisenstein and R. Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 334–343.

J. Feldman and S. Narayanan. 2004. Embodied meaning in a neural theory of language. *Brain and Language*, 89:385–392.

R. Ge and R. J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Computational Natural Language Learning (CoNLL)*, pages 9–16, Ann Arbor, Michigan.

P. Gorniak and D. Roy. 2007. Situated language understanding as filtering perceived affordances. *Cognitive Science*, 31:197–231.

T. Grenager, D. Klein, and C. D. Manning. 2005. Unsupervised learning of field segmentation models for information extraction. In *Association for Computational Linguistics (ACL)*, pages 371–378, Ann Arbor, Michigan. Association for Computational Linguistics.

R. J. Kate and R. J. Mooney. 2007. Learning language semantics from ambiguous supervision. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 895–900, Cambridge, MA. MIT Press.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *North American Association for Computational Linguistics (NAACL)*, pages 104–111, New York City. Association for Computational Linguistics.

W. Lu, H. T. Ng, W. S. Lee, and L. S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 783–792.

R. C. Moore. 2004. Improving IBM word alignment model 1. In *Association for Computational Linguistics (ACL)*, pages 518–525, Barcelona, Spain. Association for Computational Linguistics.

H. Ney and S. Vogel. 1996. HMM-based word alignment in statistical translation. In *International Conference on Computational Linguistics (COLING)*, pages 836–841. Association for Computational Linguistics.

J. M. Siskind. 1996. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61:1–38.

B. Snyder and R. Barzilay. 2007. Database-text alignment via structured multilabel classification. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1713–1718, Hyderabad, India.

C. Yu and D. H. Ballard. 2004. On the integration of grounding language and learning objects. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 488–493, Cambridge, MA. MIT Press.

L. S. Zettlemoyer and M. Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Uncertainty in Artificial Intelligence (UAI)*, pages 658–666.

L. S. Zettlemoyer and M. Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, pages 678–687.