Grounding Language with Points and Paths in Continuous Spaces

Jacob Andreas and **Dan Klein**
Computer Science Division
University of California, Berkeley
{jda,klein}@cs.berkeley.edu

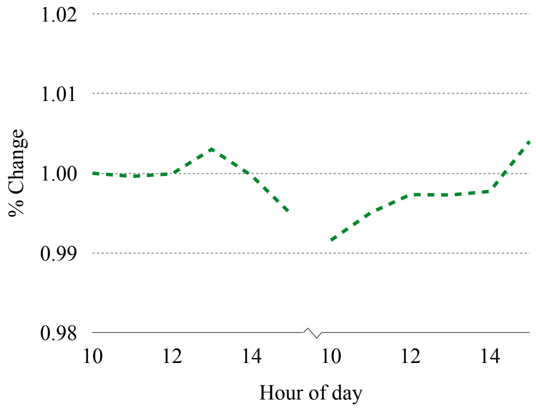
Abstract

We present a model for generating path-valued interpretations of natural language text. Our model encodes a map from natural language descriptions to paths, mediated by segmentation variables which break the language into a discrete set of events, and alignment variables which reorder those events. Within an event, lexical weights capture the contribution of each word to the aligned path segment. We demonstrate the applicability of our model on three diverse tasks: a new color description task, a new financial news task and an established direction-following task. On all three, the model outperforms strong baselines, and on a hard variant of the direction-following task it achieves results close to the state-of-the-art system described in Vogel and Jurafsky (2010).

1 Introduction

This paper introduces a probabilistic model for predicting grounded, real-valued trajectories from natural language text. A long tradition of research in compositional semantics has focused on discrete representations of meaning. The original focus of such work was on logical translation: mapping statements of natural language to a formal language like first-order logic (Zettlemoyer and Collins, 2005) or database queries (Zelle and Mooney, 1996). Subsequent work has integrated this logical translation with interpretation against a symbolic database (Liang et al., 2013).

There has been a recent increase in interest in perceptual grounding, where lexical semantics anchor in perceptual variables (points, distances, etc.) derived from images or video. Bruni et al. (2014) describe a procedure for constructing word representations using text- and image-based distributional information. Yu and Siskind (2013) describe a model for identifying scenes given descriptions, and Golland et al. (2010), Kollar et al. (2010), and Krishnamurthy and Kollar (2013) describe models for identifying individual components of scenes described by text. These all have the form of matching problems between text and observed groundings—what has been missing so far is the ability to *generate* grounded interpretations from scratch, given only text.



U.S. stocks rebound after bruising two-day swoon

Figure 1: Example stock data. The chart displays index value over a two-day period (divided by the dotted line), while the accompanying headline describes the observed behavior.

Our work continues in the tradition of this perceptual grounding work, but makes two contributions. First, our approach is able to predict simple world states (and their evolution): for a general class of continuous domains, we produce a representation of $p(\text{world} \mid \text{text})$ that admits easy sampling and maximization. This makes it possible to produce grounded interpretations of text without reference to a pre-existing scene. Simultaneously, we extend the range of temporal phenomena that can be modeled—unlike the aforementioned spatial semantics work, we consider language that de-

Proceedings of the Eighteenth Conference on Computational Language Learning, pages 58–67,
Baltimore, Maryland USA, June 26-27 2014. ©2014 Association for Computational Linguistics

scribes time-evolving trajectories, and unlike Yu and Siskind (2013), we allow these trajectories to have event substructure, and model temporal ordering. Our class of models generalizes to a variety of different domains: a new color-picking task, a new financial news task, and a more challenging variant of the direction-following task established by Vogel and Jurafsky (2010).

As an example of the kinds of phenomena we want to model, consider Figure 1, which shows the value of the Dow Jones Industrial Average over June 3rd and 4th 2008, along with a financial news headline from June 4th. There are several effects of interest here. One phenomenon we want to capture is that the lexical semantics of individual words must be combined: *swoon* roughly describes a drop while *bruising* indicates that the drop was severe. We isolate this lexical combination in Section 4, where we consider a limited model of color descriptions (Figure 2). A second phenomenon is that the description is composed of two separate events, a *swoon* and a *rebound*; moreover, those events do not occur in their textual order, as revealed by *after*. In Section 5, we extend the model to include segmentation and ordering variables and apply it to this stock data.

The situation where language describes a path through some continuous space—literal or metaphorical—is more general than stock headlines. Our claim is that a variety of problems in language share these same characteristics. To demonstrate generality of the model, we also apply it in Section 6 to a challenging variant of the direction-following task described by Vogel and Jurafsky (2010) (Figure 3), where we achieve results close to a state-of-the-art system that makes stronger assumptions about the task.

2 Three tasks in grounded semantics

The problem of inferring a structured state representation from sensory input is a hard one, but we can begin to tackle grounded semantics by restricting ourselves to cases where we have sequences of real-valued observations directly described by text. In this paper we'll consider the problems of recognizing colors, describing time series, and following navigational instructions. While these tasks have been independently studied, we believe that this is the first work which presents them in a unified framework, and carries them out with a single family of models.

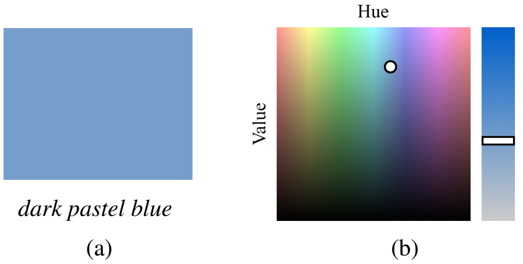


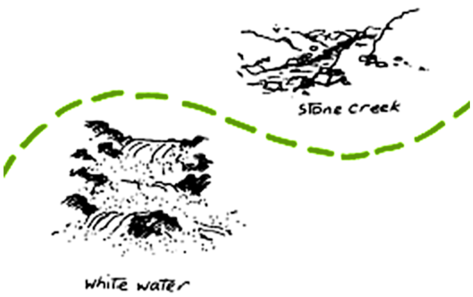
Figure 2: Example color data: (a) a named color; (b) its coordinates in color space.

Colors Figure 2 shows a color called *dark pastel blue*. English speakers, even if unfamiliar with the specific color, can identify roughly what the name signifies because of prior knowledge of the meanings of the individual words.

Because the color domain exhibits lexical compositionality but not event structure, we present it here to isolate the non-temporal compositional effects in our model. Any color visible to the human eye can be identified with three coordinates, which we'll take to be hue, saturation and value (HSV). As can be seen in Figure 2 the "hue" axis corresponds to the differentiation made by basic color names in most languages. Other modifiers act on the saturation and value axes: either simple ones like *dark* (which decreases value), or more complicated ones like *pastel* (which increases value and decreases saturation). Given a set of named colors and their HSV coordinates, a learning algorithm should be able to identify the effects of each word in the vocabulary and predict the appearance of new colors with previously-unseen combinations of modifiers.

Compositional interpretations of color have received attention in linguistics and philosophy of language (Kennedy and McNally, 2010), but while work in grounded computational semantics like that of Krishnamurthy and Kollar (2013) has succeeded in learning simple color predicates, our model is the first to capture the machine learning of color in a fine-grained, compositional way.

Time series As a first step into temporal structure, we'll consider language describing the behavior of stock market indices. Here, again, there is a simple parameterization—in this case just a single number describing the total value of the index—but as shown by the headline example in Figure 1, the language used to describe changes in the stock market can be quite complex. Head-



right round the white water [. . .] but stay quite close 'cause you don't otherwise you're going to be in that stone creek

Figure 3: Example map data: a portion of a map, and a single line from a dialog which describes navigation relative to the two visible landmarks.

lines may describe multiple events, or multi-part events like *rebound* or *extend*; stocks do not simply *rise* or *fall*, but *stagger*, *stumble*, *swoon*, and so on. There are compositional effects here as well: distinction is made between *falling* and *falling sharply*; gradual trends are distinguished from those which occur suddenly, at the beginning or end of the trading day. Along with temporal structure, the problem requires a more sophisticated treatment of *syntax* than the colors case—now we have to identify which subspans of the sentence are associated with each event observed, and determine the correspondence between surface order and actual order in time.

The learning of correspondences between text and time series has attracted more interest in natural language generation than in semantics (Yu et al., 2007). Research on natural language processing and stock data, meanwhile, has largely focused on prediction of future events (Kogan et al., 2009).

Direction following We'll conclude by applying our model to the well-studied problem of following navigational directions. A variety of reinforcement-learning approaches for following directions on a map were previously investigated by Vogel and Jurafsky (2010) using a corpus assembled by Anderson et al. (1991). An example portion of a path and its accompanying instruction is shown in Figure 3. While also representable as a set of real valued coordinates, here 2-d, this data set looks very different—a typical example consists of more than a hundred sentences of the kind shown in Figure 3, accompanying a long path. The language, a transcript of a spoken dialog, is also considerably less formal than the language found in the *Wall Street Journal* examples, involving disfluency, redundancy and occasionally errors. Nevertheless the underlying structure of this problem and the stock problem are fundamentally similar.

In addition to Vogel and Jurafsky, Tellex et al. (2011) give a weakly-supervised model for mapping single sentences to commands, and Branavan et al. (2009) give an alternative reinforcement-learning approach for following long command sequences. An intermediate between this approach and ours is the work of Chen and Mooney (2011) and Artzi and Zettlemoyer (2013), which bootstrap a semantic parser to generate logical forms specifying the output path, rather than predicting the path directly.

Between them, these tasks span a wide range of linguistic phenomena relevant to grounded semantics, and provide a demonstration of the usefulness and general applicability of our model. While development of the perceptual groundwork necessary to generalize these results to more complex state spaces remains a major problem, our three examples provide a starting point for studying the relationship between perception, time and the semantics of natural language.

3 Preliminaries

In the experiments that follow, each training example will consist of:

- Natural language text, consisting of a constituency parse tree or trees. For a given example, we will denote the associated trees $(\mathcal{T}_1, \mathcal{T}_2, \ldots)$. These are also observed at test time, and used to predict new groundings.
- A vector-valued, grounded observation, or a sequence of observations (a path), which we will denote $\mathcal{V}$ for a given example. We will further assume that each of these paths has been pre-segmented (discussed in detail in Section 5) into a sequence $(\mathcal{V}_1, \mathcal{V}_2, \ldots)$. These are only observed during training.

The probabilistic backbone of our model is a collection of linear and log-linear predictors. Thus it will be useful to work with vector-valued representations of both the language and the path, which we accomplish with a pair of feature functions ϕ_t and ϕ_v. As the model is defined only in terms of these linear representations, we can

$\phi_t(T)$	▪ Label at root of T ▪ Lemmatized leaves of T
$\phi_v(V)$	▪ Last element of V ▪ Curvature of quadratic approx. to V (stocks only)
$\phi_a(T, A_i, A_{i-1})$	Cartesian prod. of $\phi_t(T)$ with: ▪ $\mathbb{I}[A_i$ is aligned$]$ ▪ $\mathbb{I}[A_{i-1}$ is aligned$]$ ▪ $A_1 - A_{i-1}$ (if both aligned)

Table 1: Features used for linear parameterization of the grounding model.

simplify notation by writing $T_i = \phi_t(\mathcal{T}_i)$ and $V_i = \phi_v(\mathcal{V}_i)$. As the ultimate prediction task is to produce paths, and not their featurized representations, we will assume that it is also straightforward to compute ϕ_v^{-1}, which projects path features back into the original grounding domain.

All parse trees are predicted from input text using the Berkeley Parser (Petrov and Klein, 2007). Feature representations for both trees and paths are simple and largely domain-independent; they are explicitly enumerated in Table 1.

The general framework presented here leaves one significant problem unaddressed: given a large state vector encoding properties of multiple objects, how do we resolve an utterance about a single object to the correct subset of indices in the vector? While none of the tasks considered in this paper require an argument resolution step of this kind, interpretation of noun phrases is one of the better-studied problems in compositional semantics (Zelle and Mooney (1996), inter alia), and we expect generalization of this approach to be straightforward using these tools.

We will consider the color, stock, and navigation tasks in turn. It is possible to view the models we give for all three as instantiations of the same graphical model, but for ease of presentation we will introduce this model incrementally.

4 Predicting vectors

Prediction of a color variable from text has the form of a regression problem: given a vector of lexical features extracted from the name, we wish to predict the entries of a vector in color space. It seems linguistically plausible that this regression is *sparse* and *linear*: that most words, if they provide any constraints at all, tend to express preferences about a subset of the available dimensions; and that composition within the domain of a single event largely consists of words additively predicting that event's parameters, without complex nonlinear interactions. This is motivated by the observation that pragmatic concerns force linguistic descriptors to orient themselves along a small set of perceptual bases: once we have words for *north* and *east*, we tend to describe intermediates as *northeast* rather than inventing an additional word which means "a little of both".

As discussed above, we can represent a color as a point in a three-dimensional HSV space. Let T denote features on the parse tree of the color name, and V its representation in color space (consistent with the definition of ϕ_v given in Table 1). Linearity suggests the following model:

$$p(T, V) \propto e^{-\left\|\theta_t^\top T - V\right\|_2^2} \tag{1}$$

The learning problem is then:

$$\underset{\theta_t}{\operatorname{argmin}} \sum_{T,V} \left\|\theta_t^\top T - V\right\|_2^2 \tag{2}$$

which, with a sparse prior on θ_t, is the probabilistic formulation of Lasso regression (Tibshirani, 1996), for which standard tools are available in the optimization literature.

To predict color space values from a new (featurized) name T, we output:

$$\underset{V}{\operatorname{argmax}}\ p(T, V) = \theta_t^\top T$$

4.1 Evaluation

We collect a set of color names and their corresponding HSV triples from the English Wikipedia's *List of Colors*, retaining only those color names in which every word appears at least three times in the training corpus. This leaves a set of 419 colors, which we randomly divide into a 377-item training set and 42-item test set. The model's goal will be to learn to identify new colors given only their names.

We consider two evaluations: one which measures the model's ability to distinguish the named color from a random alternative—analogous to the evaluation in Yu and Siskind (2013)—and one which measures the absolute difference between predicted and true color values. In particular, in the first evaluation the model is presented with the name of a color and a pair of candidates, one

Method	Sel. ↑	H ↓	S ↓	V ↓
Random	0.50	0.30	0.38	0.39
Last word	0.78	**0.05**	0.26	0.17
Full model	**0.81**	0.07	**0.21**	**0.13**
Human	0.86	-	-	-

Table 2: Results for the color selection task. Sel(ection accuracy) is frequency with which the system was able to correctly identify the color described when paired with a random alternative. Other columns are the magnitude of the average prediction error along the axes of the color space. Full model selection accuracy is a statistically significant ($p < 0.05$) improvement over the baseline using a paired sign test.

the color corresponding to the name and another drawn randomly from the test set, and report the fraction of times the true color is assigned a higher probability than the random alternative. In the second, we report the absolute value of the difference between true and predicted hue, saturation, and luminosity.

We compare against two baselines: one which looks only at the last word in the color name (almost always a hue category), and so captures no compositional effects, and another which outputs random values for all three coordinates. Results are shown in Table 2. The model with all lexical features outperforms both baselines on selection and all but one absolute error metric.

4.2 Error analysis

An informal experiment in which the color selection task was repeated on one of the authors' colleagues (the "Human" row in Table 2) yielded an accuracy of 86%, only 5% better than the system. While not intended as a rigorous upper bound on performance, this suggests that the model capacity and training data are sufficient to capture most interesting color behavior. The errors that do occur appear to mostly be of two kinds. In one case, a base color is seen only with a small (or related) set of modifiers, from which the system is unable to infer the meaning of the base color (e.g. from *Japanese indigo*, *lavender indigo*, and *electric indigo*, the learning algorithm infers that indigo is bright purple). In the other, no part of the color word is seen in training, and the system outputs an unrelated "default" color (*teal* is predicted to be bright red).

5 Predicting paths

The idea that a sentence's meaning is fundamentally described by a set of *events*, each associated with a set of predicates, is well-developed in neo-Davidsonian formal semantics (Parsons, 1990). We adopt the skeleton of this formal approach by tying our model to (latent) partitions of the input sentence into disjoint events. Rather than attempting to pass through a symbolic meaning representation, however, this event structure will be used to map text directly into the grounding domain. We assume that this domain has pre-existing structure—in particular, that in our input paths $\mathcal{V}$, the boundaries of events have already been identified, and that the problem of aligning text to portions of the segment only requires aligning to segment indices rather than fine-grained time indices. This is a strong assumption, and one that future work may wish to revisit, but there exist both computational tools from the changepoint detection literature (Basseville and Nikiforov, 1995) and pieces of evidence from cognitive science (Zacks and Swallow, 2007) which suggest that assuming a pre-linguistic structuring of events is a reasonable starting point.

In the text domain, we make the corresponding assumption that each of these events is *syntactically local*—that a given span of the input sentence provides information about at most one of these segmented events.

The main structural difference between the color example in Figure 2 and the stock market example in Figure 1 is the introduction of a time dimension orthogonal to the dimensions of the state space. To accommodate this change, we extend the model described in the previous subsection in the following way: Instead of a single vector, each tree representation T is paired with a *sequence* of path features $\mathbf{V} = (V_1, V_2, \ldots, V_M)$. For the time being we continue to assume that there is only one input tree per training example. As before, we wish to model the probability $p(T, \mathbf{V})$, but the problem becomes harder: a single sentence might describe multiple events, but we don't know what the correspondence is between regions of the sentence and segments $\mathbf{V}$.

Though the ultimate goal is still prediction of V vectors from novel T instances, we cannot do this without also inferring a set of latent *alignments* between portions of the path and input sentence during training. To allow a sentence to explain mul-

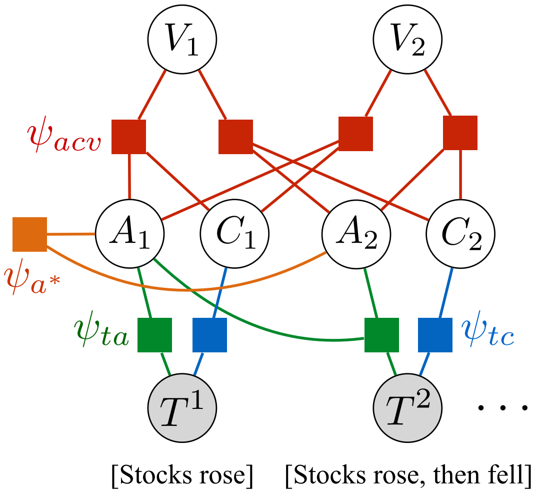


Figure 4: Factor graph for stocks grounding model. Only a subset of the alignment candidates are shown. ψ_{tc} maps text to constraints, ψ_{acv} maps constraints to grounded segments, and ψ_{ta} determines which constraints act on which segments.

tiple events, we'll break each T apart into a set of *alignment candidates* T^i. We'll allow as an alignment candidate any subtree of T, and additionally any subtree from which a single constituent has been deleted.

We then introduce two groups of latent variables: alignment variables $\mathbf{A} = (A_1, A_2, \ldots)$, which together describe a mapping from pieces of the input sentence to segments of the observed path, and what we'll call "constraint" variables $\mathbf{C} = (C_1, C_2, \ldots)$, which express each aligned tree segment's prediction about what its corresponding path should look like (so that the possibly-numerous parts of the tree aligned to a single segment can independently express preferences about the segment's path features).

In addition to ensuring that the alignment is consistent with the bracketing of the tree, it might be desirable to impose additional global constraints on the alignment. There are various ways to do this in a graphical modeling framework; the most straightforward is to add a combinatorial factor touching all alignment variables which checks for satisfaction of the global constraint. In general this makes alignment intractable. If the total number of alignments licensed by this combinatorial factor is small (i.e. if acceptable alignments are sparse within the exponentially-large set of all possible assignments to $\mathbf{A}$), it is possible to directly sum them out during inference. Otherwise approximate techniques (as discussed in the following section) will be necessary.

As discussed in Section 2, our financial timelines cover two-day periods, and it seems natural to treat each day as a separate event. Then the simple regression model described in the preceding section, extended to include alignment and constraint variables, has the form of the factor graph shown in Figure 4. In particular, the joint distribution $p(T, \mathbf{V})$ is the product of four groups of factors:

Alignment factors ψ_{ta}, which use a log-linear model to score neighboring pairs of factors with a feature function ϕ_a:

$$\psi_{ta}(T^i, A_i, A_{i-1}) = \frac{e^{\theta_a^\top \phi_a(T_i, A_i, A_{i-1})}}{\sum_{A'_i, A'_{i-1}} e^{\theta_a^\top \phi_a(T^i, A'_i, A'_{i-1})}} \tag{3}$$

Constraint factors ψ_{tc}, which map text features onto constraint values:

$$\psi_{tc}(T^i, C_i) = e^{-||\theta_t^\top T_i - C_i||_2^2} \tag{4}$$

Prediction factors ψ_{acv} which encourage predicted constraints and path features to agree:

$$\psi_{acv}(A_i, C_i, V_j) = \begin{cases} 1 & \text{if } A_i \neq j \\ e^{-||C_i - V_j||_2^2} & \text{o.w.} \end{cases} \tag{5}$$

A **global factor** $\psi_{a^*}(A_1, A_2, \cdots)$ which places an arbitrary combinatorial constraint on the alignment.

Note the essential similarity between Equations 1 and 4—in general, it can be shown that this factor model reduces to the regression model we gave for colors when there is only one of each T^i and V_j.

5.1 Learning

In order to make learning in the stocks domain tractable, we introduce the following global constraints on alignment: every terminal must be aligned, and two constituents cannot be aligned to the same segment. Together, these simplify learning by ensuring that the number of terms in the sum over $\mathbf{A}$ and $\mathbf{C}$ is polynomial (in fact $\mathcal{O}(n^2)$) in the length of the input sentence. We wish to find the maximum *a posteriori* estimate $p(\theta_t, \theta_a | T, \mathbf{V})$ for θ_t and θ_a, which we can do using the Expectation–Maximization algorithm. To find regression scoring weights θ_t, we have:

E step:

$$M = \mathbb{E}\left[\sum_i T^i (T^i)^\top\right] ; \; N = \mathbb{E}\left[\sum_i T^i V_{A_i}^\top\right] \quad (6)$$

M step:

$$\theta_t = M^{-1} N \quad (7)$$

To find alignment scoring weights θ_a, we must maximize:

$$\sum_i \mathbb{E}\left[\log\left(\frac{e^{\theta_a^\top \phi_a(A_i, A_{i-1}, T^i)}}{\sum_{A'_i, A'_{i-1}} e^{\theta_a^\top \phi_a(A'_i, A'_{i-1}, T^i)}}\right)\right] \quad (8)$$

which can be done using a variety of convex optimization tools; we used L-BFGS (Liu and Nocedal, 1989).

The predictive distribution $p(\mathbf{V}|T)$ can also be straightforwardly computed using the standard inference procedures for graphical models.

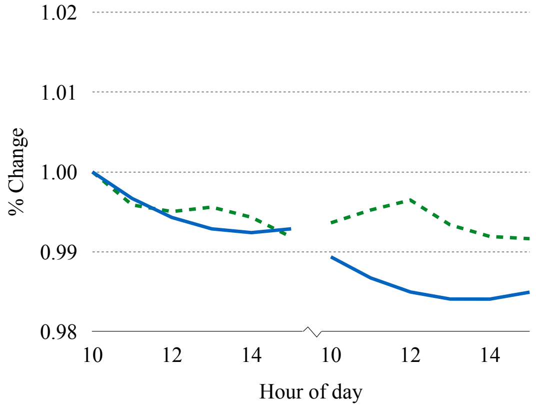
5.2 Evaluation

Our stocks dataset consists of a set of headlines from the “Market Snapshot” column of the *Wall Street Journal*’s MarketWatch website,[1] paired with hourly stock charts for each day described in a headline. Data is collected over a roughly decade-long period between 2001 and 2012; after removing weekends and days with incomplete stock data, we have a total of 2218 headline/time series pairs. As headlines most often discuss a single day or a short multi-day period, each training example consists of two days’ worth of stock data concatenated together. We use a 90%/10% train/test split, with all test examples following all training examples chronologically.

We compare against two baselines: one which uses no text (and so learns only the overall market trend during the training period), and another which uses a fixed alignment instead of summing, aligning the entire tree to the second day’s time series. Prediction error is the sum of squared errors between the predicted and gold time series.

We report both the magnitude of the prediction error, and the model’s ability to distinguish between the described path and a randomly-selected alternative. The system scores poorly on squared error (which disproportionately penalizes large deviations from the correct answer, preferring conservative models), but outperforms both baselines on the task of choosing the described stock history—when it is wrong, its errors are often large in magnitude, but its predictions more frequently resemble the correct time series than the other systems.

Figure 5 shows example system output for an example sentence. The model correctly identifies the two events, orders them in time and gets their approximate trend correct. Table 4 shows some



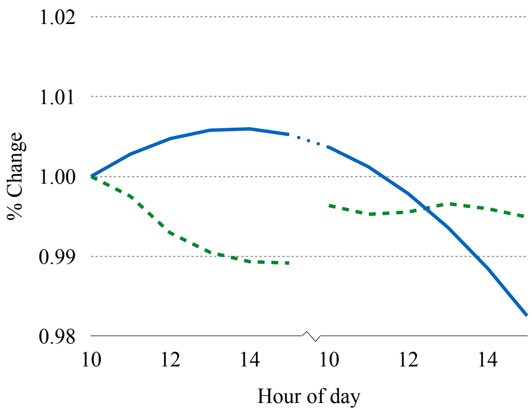
[*U.S. stocks end lower*]$_2$ [*as economic worries persist*]$_1$

Figure 5: Example output from the stocks task. The model prediction is given in blue (solid), and the reference time series in green (dashed). Brackets indicate the predicted boundaries of event-introducing spans, and subscripts their order in the sentence. The model correctly identifies that *end lower* refers to the current day, and *persist* provides information about the previous day.

Method	Sel. acc. ↑	Pred. err. ↓
No text	0.51	0.0012
Fixed alignment	0.59	**0.0011**
Full model	**0.61**	0.0018
Human	0.72	–

Table 3: Results for the stocks task. Sel(ection accuracy) measures the frequency with which the system correctly identifies the stock described in the headline when paired with a random alternative. Pred(iction error) is the mean sum of squared errors between the real and predicted paths. Full model selection accuracy is a statistically significant improvement ($p < 0.05$) over the baseline using a paired sign test.

[1] `http://www.marketwatch.com/Search?m=Column&mp=Market%20Snapshot`



[*U.S. stocks extend losing stretch*]$_1$

Figure 6: Example error from the stocks task. The system's prediction, in blue (solid), fails to segment the input into two events, and thus incorrectly extends the *losing* trend to the entire output time span.

features learned by the model—as desired, it correctly interprets a variety of different expressions used to describe stock behavior.

5.3 Error analysis

As suggested by Table 4, learned weights for the trajectory-grounded features θ_t are largely correct. Thus, most incorrect outputs from the system involve alignment to time. Many multipart events (like *rebound*) can be reasonably explained using the curvature feature without splitting the text into two segments; as a result, the system tends to be fairly conservative about segmentation and often under-segments. This results in examples like Figure 6, in which the downward trend suggested by *losing* is incorrectly extended to the entire output curve. Here, another informal experiment using humans as the predictors indicates that predictions are farther from human-level performance than they are on the colors task.

Word	Sign	Magnitude $\cdot 10^3$
rise	0.27	-0.78
swoon	-0.57	0
sharply	-0.22	0.28
slammed	-0.36	0
lifted	0.66	0

Table 4: Learned parameter settings for overall daily change, which the path featurization decomposes into a sign and a magnitude.

6 Generalizing the model

Last we consider the problem of following navigational directions. The difference between this and the previous task is largely one of scale: rather than attempting to predict the values of only two segments, we have a long string of them. The text, rather than a single tree, consists of a sequence of tens or hundreds of pre-segmented utterances.

There is one additional complication—rather than being defined in an absolute space, as they are in the case of stocks, constraints in the maps domain are provided relative to a set of known landmarks (like the *white water* and *stone creek* in Figure 3). We resolve landmarks automatically based on string matching, in a manner similar to Vogel and Jurafsky (2010), and assign each sentence in the discourse with a single referred-to landmark l_i. If no landmark is explicitly named, it inherits from the previous utterance. We continue to score constraints as before, but update the prediction factor:

$$\psi_{acv}(A_i, C_i, V_j) = \begin{cases} 1 & \text{if } A_i \neq j \\ e^{-||l_i + C_i - V_j||_2^2} & \text{o.w.} \end{cases} \quad (9)$$

The factor graph is shown in Figure 7; observe that this is simply an unrolled version of Figure 4—the basic structure of the model is unchanged. While pre-segmentation of the discourse means we can avoid aligning internal constituents of trees, we still need to treat every utterance as an alignment candidate, without a sparse combinatorial constraint. As a result, the sum over $\mathbf{A}$ and $\mathbf{C}$ is no longer tractable to compute explicitly, and approximate inference will be necessary.

For the experiments described in this paper, we do this with a sequence of Monte Carlo approximations. We run a Gibbs sampler, iteratively resampling each A_i and C_i as well as the parameter vectors θ_t and θ_a to obtain estimates of $\mathbb{E}\theta_t$ and $\mathbb{E}\theta_a$. The resampling steps for θ_t and θ_a are themselves difficult to perform exactly, so we perform an internal Metropolis-Hastings run (with a Gaussian proposal distribution) to obtain samples from the marginal distributions over θ_t and θ_a.

We approximate the mode of the posterior distribution by its mean. To follow a new set of directions in the prediction phase, we fix the parameter vectors and instead sample over $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{V}$, and output $\mathbb{E}\mathbf{V}$. To complete the prediction process

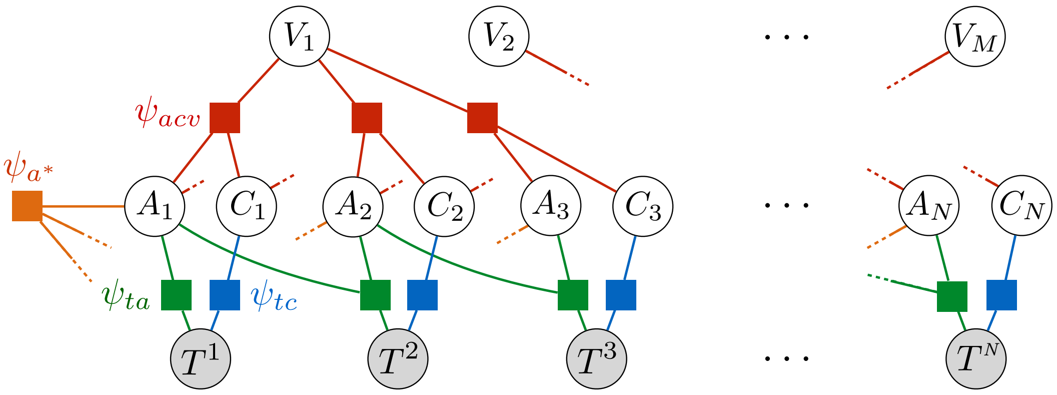


Figure 7: Factor graph for the general grounding model. Note that Figure 4 is a subgraph.

we must invert ϕ_v, which we do by producing the shortest path licensed by the features.

6.1 Evaluation

The Map Task Corpus consists of 128 dialogues describing paths on 16 maps, accompanied by transcriptions of spoken instructions, presegmented using prosodic cues. See Vogel and Jurafsky (2010) for a more detailed description of the corpus in a language learning setting. For comparability, we'll use the same evaluation as Vogel and Jurafsky, which rewards the system for moving between pairs of landmarks that also appear in the reference path, and penalizes it for additional superfluous movement. Note that we are solving a significantly harder problem: the version addressed by Vogel and Jurafsky is a discrete search problem, and the system has hard-coded knowledge that all paths pass along one of the four sides of each landmark. Our system, by contrast, can navigate to any point in $\mathbb{R}^2$, and must *learn* that most paths stay close to a named landmark.

At test time, the system is given a new sequence of text instructions, and must output the corresponding path. It is scored on the fraction of correct transitions in its output path (precision), and the fraction of transitions in the gold path recovered (recall). Vogel and Jurafsky compare their system to a policy-gradient algorithm for using language to follow natural language instructions described by Branavan et al. (2009), and we present both systems for comparison.

Results are shown in Table 5. Our system substantially outperforms the policy gradient baseline of Branavan et al., and performs close (particularly with respect to transition recall) to the system of Vogel and Jurafsky, with fewer assumptions.

System	Prec.	Recall	F_1
Branavan et al. (09)	0.31	0.44	0.36
Vogel & Jurafsky (10)	0.46	0.51	0.48
This work	0.43	0.51	0.45

Table 5: Results for the navigation task. Higher is better for all of precision, recall and F_1.

6.2 Error analysis

As in the case of stocks, most of the prediction errors on this task are a result of misalignment. In particular, many of the dialogues make passing reference to already-visited landmarks, or define destinations in empty regions of the map in terms of multiple landmarks simultaneously. In each of these cases, the system is prone to directly visiting the named landmark or landmarks instead of ignoring or interpolating as necessary.

7 Conclusion

We have presented a probabilistic model for grounding natural language text in vector-valued state sequences. The model is capable of segmenting text into a series of events, ordering these events in time, and compositionally determining their internal structure. We have evaluated on a variety of new and established applications involving colors, time series and navigation, demonstrating improvements over strong baselines in all cases.

Acknowledgments

This work was partially supported by BBN under DARPA contract HR0011-12-C-0014. The first author is supported by a National Science Foundation Graduate Research Fellowship.

References

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The HCRC map task corpus. *Language and speech*, 34(4):351–366.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1(1):49–62.

Michele Basseville and Igor V Nikiforov. 1995. Detection of abrupt changes: theory and applications. *Journal of the Royal Statistical Society-Series A Statistics in Society*, 158(1):185.

SRK Branavan, Harr Chen, Luke S Zettlemoyer, and Regina Barzilay. 2009. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 82–90. Association for Computational Linguistics.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *AAAI*, volume 2.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on Empirical Methods in Natural Language Processing*, pages 410–419. Association for Computational Linguistics.

Christopher Kennedy and Louise McNally. 2010. Color, context, and compositionality. *Synthese*, 174(1):79–98.

Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. 2009. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280. Association for Computational Linguistics.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2010. Grounding verbs of motion in natural language commands to robots. In *International Symposium on Experimental Robotics*.

Jayant Krishnamurthy and Thomas Kollar. 2013. Jointly learning to parse and perceive: connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*.

Percy Liang, Michael I Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

Terence Parsons. 1990. *Events in the semantics of English*. MIT Press.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of Human Language Technologies: The 2007 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Assocation for Computational Linguistics.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *In Proceedings of the National Conference on Artificial Intelligence*.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Adam Vogel and Dan Jurafsky. 2010. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 806–814. Association for Computational Linguistics.

Haonan Yu and Jeffrey Mark Siskind. 2013. Grounded language learning from videos described with sentences. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jin Yu, Ehud Reiter, Jim Hunter, and Chris Mellish. 2007. Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering*, 13(1):25–49.

Jeffrey M Zacks and Khena M Swallow. 2007. Event segmentation. *Current Directions in Psychological Science*, 16(2):80–84.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1050–1055.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 658–666.