

# Probabilistic Domain Modelling With Contextualized Distributional Semantic Vectors

**Jackie Chi Kit Cheung**

University of Toronto  
10 King's College Rd., Room 3302  
Toronto, ON, Canada M5S 3G4  
jcheung@cs.toronto.edu

**Gerald Penn**

University of Toronto  
10 King's College Rd., Room 3302  
Toronto, ON, Canada M5S 3G4  
gpenn@cs.toronto.edu

## Abstract

Generative probabilistic models have been used for content modelling and template induction, and are typically trained on small corpora in the target domain. In contrast, vector space models of distributional semantics are trained on large corpora, but are typically applied to domain-general lexical disambiguation tasks. We introduce Distributional Semantic Hidden Markov Models, a novel variant of a hidden Markov model that integrates these two approaches by incorporating contextualized distributional semantic vectors into a generative model as observed emissions. Experiments in slot induction show that our approach yields improvements in learning coherent entity clusters in a domain. In a subsequent extrinsic evaluation, we show that these improvements are also reflected in multi-document summarization.

## 1 Introduction

Detailed domain knowledge is crucial to many NLP tasks, either as an input for language understanding, or as the goal itself, to acquire such knowledge. For example, in information extraction, a list of slots in the target domain is given to the system, and in natural language generation, content models are trained to learn the content structure of texts in the target domain for information structuring and automatic summarization.

Generative probabilistic models have been one popular approach to content modelling. An important advantage of this approach is that the structure of the model can be adapted to fit the assumptions about the structure of the domain and the nature of the end task. As this field has progressed, the formal structures that are assumed to represent a

domain have increased in complexity and become more hierarchical. Earlier work assumes a flat set of topics (Barzilay and Lee, 2004), which are expressed as states of a latent random variable in the model. Later work organizes topics into a hierarchy from general to specific (Haghighi and Vanderwende, 2009; Celikyilmaz and Hakkani-Tur, 2010). Recently, Cheung et al. (2013) formalized a domain as a set of *frames* consisting of prototypical sequences of *events*, *slots*, and *slot fillers* or *entities*, inspired by classical AI work such as Schank and Abelson's (1977) scripts. We adopt much of this terminology in this work. For example, in the CRIMINAL INVESTIGATIONS domain, there may be events such as a murder, an investigation of the crime, an arrest, and a trial. These would be indicated by *event heads* such as *kill*, *arrest*, *charge*, *plead*. Relevant slots would include VICTIM, SUSPECT, AUTHORITIES, PLEA, etc.

One problem faced by this line of work is that, by their nature, these models are typically trained on a small corpus from the target domain, on the order of hundreds of documents. The small size of the training corpus makes it difficult to estimate reliable statistics, especially for more powerful features such as higher-order N-gram features or syntactic features.

By contrast, distributional semantic models are trained on large, domain-general corpora. These methods model word meaning using the contexts in the training corpus in which the word appears. The most popular approach today is a vector space representation, in which each dimension corresponds to some context word, and the value at that dimension corresponds to the strength of the association between the context word and the target word being modelled. A notion of word similarity arises naturally from these models by comparing the similarity of the word vectors, for example by using a cosine measure. Recently, these models have been extended by considering how distribu-

tional representations can be modified depending on the specific context in which the word appears (Mitchell and Lapata, 2008, for example). Contextualization has been found to improve performance in tasks like lexical substitution and word sense disambiguation (Thater et al., 2011).

In this paper, we propose to inject contextualized distributional semantic vectors into generative probabilistic models, in order to combine their complementary strengths for domain modelling. There are a number of potential advantages that distributional semantic models offer. First, they provide domain-general representations of word meaning that cannot be reliably estimated from the small target-domain corpora on which probabilistic models are trained. Second, the contextualization process allows the semantic vectors to implicitly encode disambiguated word sense and syntactic information, without further adding to the complexity of the generative model.

Our model, the Distributional Semantic Hidden Markov Model (DSHMM), incorporates contextualized distributional semantic vectors into a generative probabilistic model as observed emissions. We demonstrate the effectiveness of our model in two domain modelling tasks. First, we apply it to slot induction on guided summarization data over five different domains. We show that our model outperforms a baseline version of our method that does not use distributional semantic vectors, as well as a recent state-of-the-art template induction method. Then, we perform an extrinsic evaluation using multi-document summarization, wherein we show that our model is able to learn event and slot topics that are appropriate to include in a summary. From a modelling perspective, these results show that probabilistic models for content modelling and template induction benefit from distributional semantics trained on a much larger corpus. From the perspective of distributional semantics, this work broadens the variety of problems to which distributional semantics can be applied, and proposes methods to perform inference in a probabilistic setting beyond geometric measures such as cosine similarity.

## 2 Related Work

Probabilistic content models were proposed by Barzilay and Lee (2004), and related models have since become popular for summarization (Fung and Ngai, 2006; Haghghi and Vanderwende,

2009), and information ordering (Elsner et al., 2007; Louis and Nenkova, 2012). Other related generative models include topic models and structured versions thereof (Blei et al., 2003; Gruber et al., 2007; Wallach, 2008). In terms of domain learning in the form of template induction, heuristic methods involving multiple clustering steps have been proposed (Filatova et al., 2006; Chambers and Jurafsky, 2011). Most recently, Cheung et al. (2013) propose PROFINDER, a probabilistic model for frame induction inspired by content models. Our work is similar in that we assume much of the same structure within a domain and consequently in the model as well (Section 3), but whereas PROFINDER focuses on finding the “correct” number of frames, events, and slots with a nonparametric method, this work focuses on integrating global knowledge in the form of distributional semantics into a probabilistic model. We adopt one of their evaluation procedures and use it to compare with PROFINDER in Section 5.

Vector space models form the basis of modern information retrieval (Salton et al., 1975), but only recently have distributional models been proposed that are compositional (Mitchell and Lapata, 2008; Clark et al., 2008; Grefenstette and Sadrzadeh, 2011, *inter alia*), or that contextualize the meaning of a word using other words in the same phrase (*co-compositionality*) (Erk and Padó, 2008; Dinu and Lapata, 2010; Thater et al., 2011). We recently showed how such models can be evaluated for their ability to support semantic inference for use in complex NLP tasks like question answering or automatic summarization (Cheung and Penn, 2012).

Combining distributional information and probabilistic models has actually been explored in previous work. Usually, an ad-hoc clustering step precedes training and is used to bias the initialization of the probabilistic model (Barzilay and Lee, 2004; Louis and Nenkova, 2012), or the clustering is interleaved with iterations of training (Fung et al., 2003). By contrast, our method better modularizes the two, and provides a principled way to train the model. More importantly, previous ad-hoc clustering methods only use distributional information derived from the target domain itself; initializing based on domain-general distributional information can be problematic because it can bias training towards a local optimum that is inappropriate for the target domain, leading to poor per-

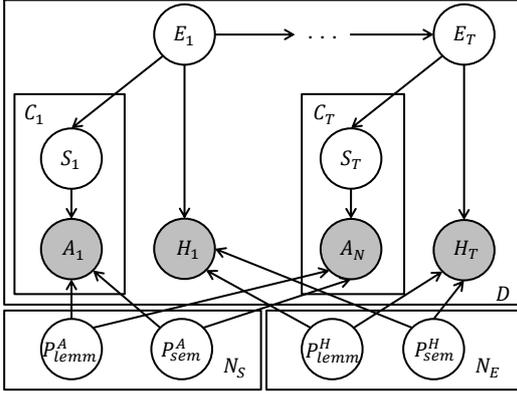


Figure 1: Graphical representation of our model. Distributions that generate the latent variables and hyperparameters are omitted for clarity.

formance.

### 3 Distributional Semantic Hidden Markov Models

We now describe the DSHMM model. This model can be thought of as an HMM with two layers of latent variables, representing events and slots in the domain. Given a document consisting of a sequence of  $T$  clauses headed by propositional heads  $\vec{H}$  (verbs or event nouns), and argument noun phrases  $\vec{A}$ , a DSHMM models the joint probability of observations  $\vec{H}$ ,  $\vec{A}$ , and latent random variables  $\vec{E}$  and  $\vec{S}$  representing domain events and slots respectively; i.e.,  $P(\vec{H}, \vec{A}, \vec{E}, \vec{S})$ .

The basic structure of our model is similar to PROFINDER. Each timestep in the model generates one clause in the document. More specifically, it generates the event heads and arguments which are crucial in identifying events and slots. We assume that event heads are verbs or event nouns, while arguments are the head words of their syntactically dependent noun phrases. We also assume that the sequence of clauses and the clause-internal syntactic structure are fixed, for example by applying a dependency parser. Within each clause, a hierarchy of latent and observed variables maps to corresponding elements in the clause (Table 1), as follows:

**Event Variables** At the top-level, a categorical latent variable  $E_t$  with  $N_E$  possible states represents the event that is described by clause  $t$ . Its value is conditioned on the previous time step’s event variable, following the standard, first-order Markov assumption ( $P^E(E_t|E_{t-1})$ , or  $P_{init}^E(E_1)$

Node	Component	Textual unit
$E_t$	Event	Clause
$S_{ta}$	Slot	Noun phrase
$H_t$	Event head	Verb/event noun
$A_{ta}$	Event argument	Noun phrase

Table 1: The correspondence between nodes in our graphical model, the domain components that they model, and the related elements in the clause.

for the first clause). The internal structure of the clause is generated by conditioning on the state of  $E_t$ , including the head of the clause, and the slots for each argument in the clause.

**Slot Variables** Categorical latent variables with  $N_S$  possible states represent the slot that an argument fills, and are conditioned on the event variable in the clause,  $E_t$  (i.e.,  $P^S(S_{ta}|E_t)$ , for the  $a$ th slot variable). The state of  $S_{ta}$  is then used to generate an argument  $A_{ta}$ .

**Head and Argument Emissions** The head of the clause  $H_t$  is conditionally dependent on  $E_t$ , and each argument  $A_{ta}$  is likewise conditioned on its slot variable  $S_{ta}$ . Unlike in most applications of HMMs in text processing, in which the representation of a token is simply its word or lemma identity, tokens in DSHMM are also associated with a vector representation of their meaning *in context* according to a distributional semantic model (Section 3.1). Thus, the emissions can be decomposed into pairs  $H_t = (\text{lemma}(H_t), \text{sem}(H_t))$  and  $A_{ta} = (\text{lemma}(A_{ta}), \text{sem}(A_{ta}))$ , where *lemma* and *sem* are functions that return the lemma identity and the semantic vector respectively. The probability of the head of a clause is thus:

$$P^H(H_t|E_t) = P_{lemm}^H(\text{lemma}(H_t)|E_t) \quad (1) \\ \times P_{sem}^H(\text{sem}(H_t)|E_t),$$

and the probability of a clausal argument is likewise:

$$P^A(A_{ta}|S_{ta}) = P_{lemm}^A(\text{lemma}(A_{ta})|S_{ta}) \quad (2) \\ \times P_{sem}^A(\text{sem}(A_{ta})|S_{ta}).$$

All categorical distributions are smoothed using add- $\delta$  smoothing (i.e., uniform Dirichlet priors). Based on the independence assumptions described above, the joint probability distribution can be fac-

tored into:

$$\begin{aligned}
P(\vec{H}, \vec{A}, \vec{E}, \vec{S}) &= P_{init}^E(E_1) \\
&\times \prod_{t=2}^T P^E(E_t|E_{t-1}) \prod_{t=1}^T P^H(H_t|E_t) \\
&\times \prod_{t=1}^T \prod_{a=1}^{C_t} P^S(S_{ta}|E_t) P^A(A_{ta}|S_{ta}).
\end{aligned} \quad (3)$$

### 3.1 Vector Space Models of Semantics

In this section, we describe several methods for producing the semantic vectors associated with each event head or argument; i.e., the function *sem*. We chose several simple, but widely studied models, to investigate whether they can be effectively integrated into DSHMM. We start with a description of the training of a basic model without any contextualization, then describe several contextualized models based on recent work.

**Simple Vector Space Model** In the basic version of the model (SIMPLE), we train a term-context matrix, where rows correspond to target words, and columns correspond to context words. Training begins by counting context words that appear within five words of the target word, ignoring stopwords. We then convert the raw counts to positive pointwise mutual information scores, which has been shown to improve word similarity correlation results (Turney and Pantel, 2010). We set thresholds on the frequencies of words for inclusion as target and context words (given in Section 4). Target words which fall below the threshold are modelled as *UNK*. All the methods below start from this basic vector representation.

**Component-wise Operators** Mitchell and Lapata (2008) investigate using component-wise operators to combine the vectors of verbs and their intransitive subjects. We use component-wise operators to contextualize our vectors, but by combining with all of the arguments, and regardless of the event head’s category. Let event head  $h$  be the syntactic head of a number of arguments  $a_1, a_2, \dots, a_m$ , and  $\vec{v}_h, \vec{v}_{a_1}, \vec{v}_{a_2}, \dots, \vec{v}_{a_m}$  be their respective vector representations according to the SIMPLE method. Then, their contextualized vectors  $\vec{c}_h^{M\&L}, \vec{c}_{a_1}^{M\&L}, \dots, \vec{c}_{a_m}^{M\&L}$  would be:

$$\vec{c}_h^{M\&L} = \vec{v}_h \odot \left( \bigodot_{i=1}^m \vec{v}_{a_i} \right) \quad (4)$$

$$\vec{c}_{a_i}^{M\&L} = \vec{v}_{a_i} \odot \vec{v}_h, \forall i = 1 \dots m, \quad (5)$$

where  $\odot$  represents a component-wise operator, addition or multiplication, and  $\bigodot$  represents its repeated application. We tested component-wise addition (M&L+) and multiplication (M&L $\times$ ).

**Selectional Preferences** Erk and Padó (2008) (E&P) incorporate *inverse selectional preferences* into their contextualization function. The intuition is that a word should be contextualized such that its vector representation becomes more similar to the vectors of other words that its dependency neighbours often take in the same syntactic position. For example, suppose *catch* is the head of the noun *ball*, in the relation of a direct object. Then, the vector for *ball* would be contextualized to become similar to the vectors for other frequent direct objects of *catch*, such as *baseball*, or *cold*. Likewise, the vector for *catch* would be contextualized to become similar to the vectors for *throw*, *hit*, etc. Formally, let  $h$  take  $a$  as its argument in relation  $r$ . Then:

$$\vec{c}_h^{E\&P} = \vec{v}_h \times \prod_{i=1}^m \sum_{w \in \mathcal{L}} freq(w, r, a_i) \cdot \vec{v}_w, \quad (6)$$

$$\vec{c}_a^{E\&P} = \vec{v}_a \times \sum_{w \in \mathcal{L}} freq(h, r, w) \cdot \vec{v}_w, \quad (7)$$

where  $freq(h, r, a)$  is the frequency of  $h$  occurring as the head of  $a$  in relation  $r$  in the training corpus,  $\mathcal{L}$  is the lexicon, and  $\times$  represents component-wise multiplication.

### Dimensionality Reduction and Vector Emission

After contextualization, we apply singular value decomposition (SVD) for dimensionality reduction to reduce the number of model parameters, keeping the  $k$  most significant singular values and vectors. In particular, we apply SVD to the  $m$ -by- $n$  term-context matrix  $M$  produced by the SIMPLE method, resulting in the truncated matrices  $M \approx U_k \Sigma_k V_k^T$ , where  $U_k$  is a  $m$ -by- $k$  matrix,  $\Sigma_k$  is  $k$ -by- $k$ , and  $V_k$  is  $n$ -by- $k$ . This takes place after contextualization, so the component-wise operators apply in the original semantic space. Afterwards, the contextualized vector in the original space,  $\vec{c}$ , can be transformed into a vector in the reduced space,  $\vec{c}^R$ , by  $\vec{c}^R = \Sigma_k^{-1} V_k^T \vec{c}$ .

Distributional semantic vectors are traditionally compared by measures which ignore vector magnitudes, such as cosine similarity, but a multivariate Gaussian is sensitive to magnitudes. Thus, the final step is to normalize  $\vec{c}^R$  into a unit vector by dividing it by its L2 norm,  $\|\vec{c}^R\|$ .

We model the emission of these contextualized vectors in DSHMM as multivariate Gaussian distributions, so the semantic vector emissions can be written as  $P_{sem}^H, P_{sem}^A \sim \mathcal{N}(\mu, \Sigma)$ , where  $\mu \in \mathbb{R}^k$  is the mean and  $\Sigma \in \mathbb{R}^{k \times k}$  is the covariance matrix. To avoid overfitting, we regularize the covariance using its conjugate prior, the Inverse-Wishart distribution. We follow the “neutral” setting of hyperparameters given by Ornoneit and Tresp (1995), so that the MAP estimate for the covariance matrix for (event or slot) state  $i$  becomes:

$$\Sigma_i = \frac{\sum_j r_{ij}(x_j - \mu_i)(x_j - \mu_i)^T + \beta I}{\sum_j r_{ij} + 1}, \quad (8)$$

where  $j$  indexes all the relevant semantic vectors  $x_j$  in the training set,  $r_{ij}$  is the posterior responsibility of state  $i$  for vector  $x_j$ , and  $\beta$  is the remaining hyperparameter that we tune to adjust the amount of regularization. To further reduce model complexity, we set the off-diagonal entries of the resulting covariance matrix to zero.

### 3.2 Training and Inference

Inference in DSHMM is accomplished by the standard Inside-Outside and tree-Viterbi algorithms, except that the tree structure is fixed, so there is no need to sum over all possible subtrees. Model parameters are learned by the Expectation-Maximization (EM) algorithm. We tune the hyperparameters  $(N_E, N_S, \delta, \beta, k)$  and the number of EM iterations by two-fold cross-validation<sup>1</sup>.

### 3.3 Summary and Generative Process

In summary, the following steps are applied to train a DSHMM:

1. Train a distributional semantic model on a large, domain-general corpus.
2. Preprocess and generate contextualized vectors of event heads and arguments in the small corpus in the target domain.
3. Train the DSHMM using the EM algorithm.

The formal generative process is as follows:

1. Draw categorical distributions  $P_{init}^E$ ;  $P^E, P^S, P_{lemm}^H$  (one per event state);  $P_{lemm}^A$  (one per slot state) from Dirichlet priors.
2. Draw multivariate Gaussians  $P_{sem}^H, P_{sem}^A$  for each event and slot state, respectively.

3. Generate the documents, clause by clause.

Generating a clause at position  $t$  consists of these steps:

1. Generate the event state  $E_t \sim P^E$  (or  $P_{init}^E$ ).
2. Generate the event head components  $lemm(H_t) \sim P_{lemm}^H, sem(H_t) \sim P_{sem}^H$ .
3. Generate a number of slot states  $S_{ta} \sim P^S$ .
4. For each slot, generate the argument components  $lemm(A_{ta}) \sim P_{lemm}^A, sem(A_{ta}) \sim P_{sem}^A$ .

## 4 Experiments

We trained the distributional semantic models using the Annotated Gigaword corpus (Napoles et al., 2012), which has been automatically preprocessed and is based on Gigaword 5th edition. This corpus contains almost ten million news articles and more than 4 billion tokens. We used those articles marked as “stories” — the vast majority of them. We modelled the 50,000 most common lemmata as target words, and the 3,000 most common lemmata as context words.

We then trained DSHMM and conducted our evaluations on the TAC 2010 guided summarization data set (Owczarzak and Dang, 2010). Lemmatization and extraction of event heads and arguments are done by preprocessing with the Stanford CoreNLP tool suite (Toutanova et al., 2003; de Marneffe et al., 2006). This data set contains 46 topic clusters of 20 articles each, grouped into five topic categories or domains. For example, one topic cluster in the ATTACK category is about the *Columbine Massacre*. Each topic cluster contains eight human-written “model” summaries (“model” here meaning a gold standard). Half of the articles and model summaries in a topic cluster are used in the guided summarization task, and the rest are used in the update summarization task.

We chose this data set because it allows us to conduct various domain-modelling evaluations. First, templates for the domains are provided, and the model summaries are annotated with slots from the template, allowing for an intrinsic evaluation of slot induction (Section 5). Second, it contains multiple domain instances for each of the domains, and each domain instance comes annotated with eight model summaries, allowing for an extrinsic evaluation of our system (Section 6).

<sup>1</sup>The topic cluster splits and the hyperparameter settings are available at <http://www.cs.toronto.edu/~jcheung/dshmm/dshmm.html>.

## 5 Guided Summarization Slot Induction

We first evaluated our models on their ability to produce coherent clusters of entities belonging to the same slot, adopting the experimental procedure of Cheung et al. (2013).

As part of the official TAC evaluation procedure, model summaries were manually segmented into *contributors*, and labelled with the slot in the TAC template that the contributor expresses. For example, a summary fragment such as *On 20 April 1999, a massacre occurred at Columbine High School* is segmented into the contributors: (*On 20 April 1999*, WHEN); (*a massacre occurred*, WHAT); and (*at Columbine High School*, WHERE).

In the slot induction evaluation, this annotation is used as follows. First, the maximal noun phrases are extracted from the contributors and clustered based on the TAC slot of the contributor. These clusters of noun phrases then become the gold standard clusters against which automatic systems are compared. Noun phrases are considered to be matched if the lemmata of their head words are the same and they are extracted from the same summary. This accounts for the fact that human annotators often only label the first occurrence of a word that belongs to a slot in a summary, and follows the standard evaluation procedure in previous information extraction tasks, such as MUC-4. Pronouns and demonstratives are ignored. This extraction process is noisy, because the meaning of some contributors depends on an entire verb phrase, but we keep this representation to allow a direct comparison to previous work.

Because we are evaluating unsupervised systems, the clusters produced by the systems are not labelled, and must be matched to the gold standard clusters. This matching is performed by mapping to each gold cluster the best system cluster according to F1. The same system cluster may be mapped multiple times, because several TAC slots can overlap. For example, in the NATURAL DISASTERS domain, an *earthquake* may fit both the WHAT slot as well as the CAUSE slot, because it generated a *tsunami*.

We trained a DSHMM separately for each of the five domains with different semantic models, tuning hyperparameters by two-fold cross-validation. We then extracted noun phrase clusters from the model summaries according to the slot labels produced by running the Viterbi algorithm on them.

Method	P	R	F1
HMM w/o semantics	13.8	64.1	22.6*
DSHMM w/ SIMPLE	20.9	27.5	23.7
DSHMM w/ E&P	20.7	27.9	23.8
PROFINDER	<b>23.7</b>	25.0	24.3
DSHMM w/ M&L+	19.7	<b>36.3</b>	25.6*
DSHMM w/ M&L×	22.1	33.2	<b>26.5*</b>

Table 2: Slot induction results on the TAC guided summarization data set. Asterisks (\*) indicate that the model is statistically significantly different from PROFINDER in terms of F1 at  $p < 0.05$ .

**Results** We compared DSHMM to two baselines. Our first baseline is PROFINDER, a state-of-the-art template inducer which Cheung et al. (2013) showed to outperform the previous heuristic clustering method of Chambers and Jurafsky (2011). Our second baseline is our DSHMM model, without the semantic vector component, (HMM w/o semantics). To calculate statistical significance, we use the paired bootstrap method, which can accommodate complex evaluation metrics like F1 (Berg-Kirkpatrick et al., 2012).

Table 2 shows that performance of the models. Overall, PROFINDER significantly outperforms the HMM baseline, but not any of the DSHMM models by F1. DSHMM with contextualized semantic vectors achieves the highest F1s, and are significantly better than PROFINDER. All of the differences in precision and recall between PROFINDER and the other models are significant. The baseline HMM model has highly imbalanced precision and recall. We think this is because the model is unable to successfully produce coherent clusters, so the best-case mapping procedure during evaluation picked large clusters that have high recall. PROFINDER has slightly higher precision, which may be due to its non-parametric split-merge heuristic. We plan to investigate whether this learning method could improve DSHMM’s performance further. Importantly, the contextualization of the vectors seems to be beneficial, at least with the M&L component-wise operators. In the next section, we show that the improvement from contextualization transfers to multi-document summarization results.

## 6 Multi-document Summarization: An Extrinsic Evaluation

We next evaluated our models extrinsically in the setting of extractive, multi-document summarization. To use the trained DSHMM for extractive summarization, we need a decoding procedure for selecting sentences in the source text to include in the summary. Inspired by the KLSUM and HIRSUM methods of Haghighi and Vanderwende (2009), we develop a criterion based on Kullback-Leibler (KL) divergence between distributions estimated from the source text, and those estimated from the summary. The assumption here is that these distributions should match in a good summary. We describe two methods to use this criterion: a basic unsupervised method (Section 6.1), and a supervised variant that makes use of in-domain summaries to learn the salient slots and events in the domain (Section 6.2).

### 6.1 A KL-based Criterion

There are four main component distributions from our model that should be considered during extraction: (1) the distribution of events, (2) the distribution of slots, (3) the distribution of event heads, and (4) the distribution of arguments. We estimate (1) as the context-independent probability of being in a certain event state, which can be calculated using the Inside-Outside algorithm. Given a collection of documents  $D$  which make up the source text, the distribution of event topics  $\hat{P}^E(E)$  is estimated as:

$$\hat{P}^E(E = e) = \frac{1}{Z} \sum_{d \in D} \sum_t \frac{In_t(e)Out_t(e)}{P(d)}, \quad (9)$$

where  $In_t(e)$  and  $Out_t(e)$  are the values of the inside and outside trellises at timestep  $t$  for some event state  $e$ , and  $Z$  is a normalization constant. The distribution for a set of sentences in a candidate summary,  $\hat{Q}^E(E)$ , is identical, except the summation is over the clauses in the candidate summary. Slot distributions  $\hat{P}^S(S)$  and  $\hat{Q}^S(S)$  (2) are defined analogously, where the summation occurs along all the slot variables.

For (3) and (4), we simply use the MLE estimates of the lemma emissions, where the estimates are made over the source text and the candidate summary instead of over the entire training set. All of the candidate summary distributions (i.e., the “ $\hat{Q}$  distributions”) are smoothed by

a small amount, so that the KL-divergence is always finite. Our KL criterion combines the above components linearly, weighting the lemma distributions by the probability of their respective event or slot state:

$$\begin{aligned} KLScore = & \quad (10) \\ & \mathcal{D}_{KL}(\hat{P}^E || \hat{Q}^E) + \mathcal{D}_{KL}(\hat{P}^S || \hat{Q}^S) \\ & + \sum_{e=1}^{N_E} \hat{P}^E(e) \mathcal{D}_{KL}(\hat{P}^H(H|e) || \hat{Q}^H(H|e)) \\ & + \sum_{s=1}^{N_S} \hat{P}^S(s) \mathcal{D}_{KL}(\hat{P}^A(A|s) || \hat{Q}^A(A|s)) \end{aligned}$$

To produce a summary, sentences from the source text are greedily added such that  $KLScore$  is minimized at each step, until the desired summary length is reached, discarding sentences with fewer than five words.

### 6.2 Supervised Learning

The above unsupervised method results in summaries that closely mirror the source text in terms of the event and slot distributions, but this ignores the fact that not all such topics should be included in a summary. It also ignores genre-specific, stylistic considerations about characteristics of good summary sentences. For example, Woodsend and Lapata (2012) find several factors that indicate sentences should *not* be included in an extractive summary, such as the presence of personal pronouns. Thus, we implemented a second method, in which we modify the KL criterion above by estimating  $\hat{P}^E$  and  $\hat{P}^S$  from other model summaries that are drawn from the same domain (i.e. topic category), except for those summaries that are written for the specific topic cluster to be used for evaluation.

### 6.3 Method and Results

We used the best performing models from the slot induction task and the above unsupervised and supervised methods based on KL-divergence to produce 100-word summaries of the guided summarization source text clusters. We did not compare against PROFINDER, as its structure is different and would have required a different procedure than the KL-criterion we developed above. As shown in the previous evaluation, however, the HMM baseline without semantics and DSHMM with SIMPLE perform similarly in terms of F1,

Method	ROUGE-1		ROUGE-2		ROUGE-SU4	
	<i>unsup.</i>	<i>sup.</i>	<i>unsup.</i>	<i>sup.</i>	<i>unsup.</i>	<i>sup.</i>
Leading baseline	28.0	–	5.39	–	8.6	–
HMM w/o semantics	32.3	32.7	<b>6.45</b>	6.49	10.1	10.2
DSHMM w/ SIMPLE	32.1	32.7	5.81	6.50	9.8	10.2
DSHMM w/ M&L+	32.1	33.4	6.27	6.82	10.0	10.6
DSHMM w/ M&L×	32.4	<b>34.3*</b>	6.35	7.11 <sup>^</sup>	10.2	<b>11.0*</b>
DSHMM w/ E&P	<b>32.8</b>	33.8*	6.38	<b>7.31*</b>	<b>10.3</b>	10.8*

Table 3: TAC 2010 summarization results by three settings of ROUGE. Asterisks (\*) indicate that the model is statistically significantly better than the HMM model without semantics at a 95% confidence interval, a caret ^ indicates that the value is marginally so.

so we consider these competitive baselines. We did not evaluate with the update summarization task, because our method has not been adapted to it. For the evaluation measure, we used the standard ROUGE suite of automatic evaluation measures (Lin, 2004). Note that the evaluation conditions of TAC 2010 are different, and thus those results are not directly comparable to ours. For instance, top performing systems in TAC 2010 make use of manually constructed lists of entities known to fit the slots in the provided templates and sample topic statements, which our method automatically learns. We include the leading baseline results from the competition as a point of reference, as it is a well-known and non-trivial one for news articles. This baseline summary consists of the leading sentences from the most recent document in the source text cluster up to the word length limit.

Table 3 shows the summarization results for the three most widely-used settings of ROUGE. All of our models outperform the leading baseline by a large margin, demonstrating the effective of the KL-criterion. In terms of unsupervised performance, all of our models perform similarly. Because the unsupervised method mimics the distributions in the source text at all levels, the method may negate the benefit of learning and simply produce summaries that match the source text in the word distributions, thus being an approximation of KLSUM. Looking at the supervised results, however, the semantic vector models show clear gains in ROUGE, whereas the baseline method does not obtain much benefit from supervision. As in the previous evaluation, the models with contextualized semantic vectors provide the best performance. M&L× performs very well, as in slot induction, but E&P also performs well, unlike in

the previous evaluation. This result reinforces the importance of the contextualization procedure for distributional semantic models.

**Analysis** To better understand what is gained by supervision using in-domain summaries, we analyzed the best performing M&L× model’s output summaries for one document cluster from each domain. For each event state, we calculated the ratio  $\hat{P}_{summ}^E(e)/\hat{P}_{source}^E(e)$ , for the probability of an event state  $e$  as estimated from the training summaries and the the source text respectively. Likewise, we calculated  $\hat{P}_{summ}^S(s)/\hat{P}_{source}^S(s)$  for the slot states. This ratio indicates the change in state’s probability after supervision; the greater the ratio, the more preferred that state becomes after training. We selected the most preferred and dis-preferred event and slot for each document cluster, and took the three most probable lemmata from the associated lemma distribution (Table 4). It seems that supervision is beneficial because it picks out important event heads and arguments in the domain, such as *charge*, *trial*, and *murder* in the TRIALS domain. It also helps the summarizer avoid semantically generic words (*be* or *have*), pronouns, quotatives, and common but irrelevant words (*home*, *city*, *restaurant* in TRIALS).

## 7 Conclusion

We have shown that contextualized distributional semantic vectors can be successfully integrated into a generative probabilistic model for domain modelling, as demonstrated by improvements in slot induction and multi-document summarization. The effectiveness of our model stems from the use of a large domain-general corpus to train the distributional semantic vectors, and the implicit syntactic and word sense information pro-

Domain	Event Heads		Slot Arguments	
	+	-	+	-
ATTACKS	<i>say</i> <sup>2</sup> , <i>cause</i> , <i>doctor</i>	<i>say</i> <sup>2</sup> , <i>be</i> , <i>have</i>	<i>attack</i> , <i>hostage</i> , <i>troops</i>	<i>he</i> , <i>it</i> , <i>they</i>
TRIALS	<i>charge</i> , <i>trial</i> , <i>accuse</i>	<i>say</i> , <i>be</i> , <i>have</i>	<i>prison</i> , <i>murder</i> , <i>charge</i>	<i>home</i> , <i>city</i> , <i>restau- rant</i>
RESOURCES	<i>reduce</i> , <i>increase</i> , <i>university</i>	<i>say</i> , <i>be</i> , <i>have</i>	<i>government</i> , <i>effort</i> , <i>program</i>	<i>he</i> , <i>they</i> , <i>it</i>
DISASTERS	<i>flood</i> , <i>strengthen</i> , <i>engulf</i>	<i>say</i> , <i>be</i> , <i>have</i>	<i>production</i> , <i>statoil</i> , <i>barrel</i>	<i>he</i> , <i>it</i> , <i>they</i>
HEALTH	<i>be</i> , <i>department</i> , <i>have</i>	<i>say</i> , <i>do</i> , <i>make</i>	<i>food</i> , <i>product</i> , <i>meat</i>	<i>she</i> , <i>people</i> , <i>way</i>

Table 4: Analysis of the most probable event heads and arguments in the most preferred (+) and dispreferred (-) events and slots after supervised training.

vided by the contextualization process. Our approach is modular, and allows principled training of the probabilistic model using standard techniques. While we have focused on the overall clustering of entities and the distribution of event and slot topics in this work, we would also like to investigate discourse modelling and content structuring. Finally, our work shows that the application of distributional semantics to NLP tasks need not be confined to lexical disambiguation. We would like to see modern distributional semantic methods incorporated into an even greater variety of applications.

## Acknowledgments

This work is supported by the Natural Sciences and Engineering Research Council of Canada.

## References

- Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.
- Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea, July. Association for Computational Linguistics.

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3.

- Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824, Uppsala, Sweden, July. Association for Computational Linguistics.

- Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA, June. Association for Computational Linguistics.

- Jackie Chi Kit Cheung and Gerald Penn. 2012. Evaluating distributional models of semantics for syntactically invariant inference. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–43, Avignon, France, April. Association for Computational Linguistics.

- Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

- Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*.

- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

- Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceed-*

<sup>2</sup>The event head *say* happens to appear in both the most preferred and dispreferred events in the ATTACKS domain.

- ings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1162–1172.
- Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, Rochester, New York, April. Association for Computational Linguistics.
- Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906.
- Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 207–214, Sydney, Australia, July. Association for Computational Linguistics.
- Pascale Fung and Grace Ngai. 2006. One story, one flow: Hidden markov story models for multilingual multidocument summarization. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2):1–16.
- Pascale Fung, Grace Ngai, and Chi-Shun Cheung. 2003. Combining optimal clustering and hidden markov models for extractive summarization. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 21–28, Sapporo, Japan, July. Association for Computational Linguistics.
- Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Amit Gruber, Michael Rosen-Zvi, and Yair Weiss. 2007. Hidden topic markov models. *Artificial Intelligence and Statistics (AISTATS)*.
- Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado, June. Association for Computational Linguistics.
- Chin Y. Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Stan Szpakowicz and Marie-Francine Moens, editors, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.
- Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea, July. Association for Computational Linguistics.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244.
1992. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann.
- Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the NAACL-HLT Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100.
- Dirk Ormoneit and Volker Tresp. 1995. Improved gaussian mixture density estimates using bayesian penalty terms and network averaging. In *Advances in Neural Information Processing*, pages 542–548.
- Karolina Owczarzak and Hoa T. Dang. 2010. TAC 2010 guided summarization task guidelines.
- Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.
- Roger C. Schank and Robert P. Abelson. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. Lawrence Erlbaum, July.
- Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word meaning in context: A simple and effective vector model. In *Proceedings of IJCNLP*.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, page 180.
- Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.
- Hanna M. Wallach. 2008. Structured topic models for language. *Doctoral dissertation, University of Cambridge*.
- Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea, July. Association for Computational Linguistics.