TRAINING A STATISTICAL SURFACE REALISER FROM AUTOMATIC SLOT LABELLING

Heriberto Cuayáhuitl, Nina Dethlefs, Helen Hastie, Xingkun Liu

School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, United Kingdom
{h.cuayahuitl, n.s.dethlefs, h.hastie, x.liu}@hw.ac.uk

ABSTRACT

Training a statistical surface realiser typically relies on labelled training data or parallel data sets, such as corpora of paraphrases. The procedure for obtaining such data for new domains is not only time-consuming, but it also restricts the incorporation of new semantic slots during an interaction, i.e. using an online learning scenario for automatically extended domains. Here, we present an alternative approach to statistical surface realisation from unlabelled data through automatic semantic slot labelling. The essence of our algorithm is to cluster clauses based on a similarity function that combines lexical and semantic information. Annotations need to be reliable enough to be utilised within a spoken dialogue system. We compare different similarity functions and evaluate our surface realiser—trained from unlabelled data—in a human rating study. Results confirm that a surface realiser trained from automatic slot labels can lead to outputs of comparable quality to outputs trained from human-labelled inputs.

Index Terms— dialogue systems, semantic slot labelling, surface realisation, unsupervised and supervised learning.

1. INTRODUCTION

Many natural language processing modules are trained on a set of labelled examples and are then ideally expected to also work on unseen inputs. However, while POS taggers or parsers can often return reasonable analyses for unseen inputs, this is not equally true for surface realisation within Spoken Dialogue Systems (SDS). A surface realiser within an SDS typically receives a single dialogue act as input [1], which lacks the detailed syntactic and lexical annotation which surface realisers from other domains can fall back on [2]. Consequently, such kinds of realisers will fail when suddenly presented with an unseen input dialogue act, restricting them to predictable and pre-trainable domains.

Our scenario is surface realisation of *known* and *unknown* dialogue acts within a spoken dialogue system in the restaurant domain. A dialogue act is defined by a dialogue act type and slot-value pairs[1]. The assumption is that the system can recognise new slots in user queries (such as a user asking about *child-friendly* restaurants when this slot was not in the training data). The system is then able to retrieve the new information from the web and dynamically extend its domain ontology. Correspondingly, the surface realiser needs to be re-trained from unlabelled data to produce meaningful output. The latter will be the focus of this paper.

Several authors have recognised the need to move away from methods that require human annotations. Instead, recent work has explored alternative techniques that require less supervision. This includes learning from trial and error [3, 4] in situated scenarios, or learning from parallel corpora [5, 6], databases [7, 8] or automatic slot filling [9]. While trial and error learning is usually not an option in scenarios involving interaction with humans–because of the high number of training episodes needed–learning from parallel corpora or databases is restricted to domains for which such resources exist. Therefore, while all of these approaches represent important steps towards training surface realisers with less supervision, they are often not readily transferable across domains. Here, we present an alternative approach to training a surface realiser from unlabelled domain data. In essence, our approach relies on the automatic assignment of semantic slot types to unlabelled data. To this end, we present an algorithm that takes as input a set of unlabelled sentences (which can consist of one or more clauses), and returns a set of semantically annotated clauses that can serve as training data for our surface realiser. The algorithm clusters unlabelled input clauses based on their similarity estimated as a function of lexical and semantic similarity. The crux of our method is therefore to find a suitable similarity metric that allows the estimation of semantically meaningful clusters. Based on these clusters, clauses can be chunked into phrases and slot values can be identified. In an automatic evaluation, both clustering and slot value detection obtain accuracies of over 90%. In addition, a human rating study confirms that a statistical surface realiser trained from unlabelled data can achieve similar performance to one trained from labelled data.

This research was funded by the European Commission FP7 programme FP7/2011-14 under grant agreement no. 287615 (PARLANCE).

[1] We use the term 'slot' and 'slot type' interchangeably, though 'attribute type' is also used in the literature. A slot type represents a set of slot values, e.g. the slot type *foodtype* has slot values {*chinese, japanese, mexican, ...*}.

2. RELATED WORK

Recent years have seen a surge of interest in unsupervised or weakly supervised methods that learn semantic concepts or linguistic expressions (for natural language generation or understanding) from unlabelled data, and move towards replacing the expensive/impractical labelled corpora required in supervised learning algorithms.

Two popular approaches to do this have been to use (a) parallel corpora [5, 6] or (b) databases [7, 8] instead of annotations. For example, [10] and [11] induce semantic parsers from parallel corpora that contain pairs of semantic forms and natural language realisations. [1] train a dynamic Bayesian network from semantically aligned data (a mapping from dialogue acts to surface realisations) produced by human annotators. Other approaches follow the same approach for natural language generation [6, 12, 13, 14]. [15] compare Hidden Markov models and Bayesian networks for statistical surface realisation from manually annotated data for a wayfinding interactive system. In essence, these learning approaches reduce the problem of automatic language induction to finding a mapping between two alternative abstract representations.

A separate direction has been to learn language through observation or trial-and-error search in situated scenarios, such as route direction generation. Methods explored here include learning from experience [16, 17], learning through observation of human behaviour [18], and learning from trial and error [3, 4]. These approaches receive their supervision from the real world and are therefore often only transferable to generation contexts using simulation [19, 20, 21]. This is due to the large number of examples required to train a surface realisation module and the (inappropriate) amount of feedback that would be required from human raters.

This paper goes beyond previous work in statistical surface realisation. To the best of our knowledge, our method—based on an almost unsupervised learning approach—is the first to address the problem of training a statistical surface realiser from unlabelled, automatically annotated, data.

3. METHOD FOR SEMANTIC SLOT LABELLING

The intuition behind our approach to semantic slot labelling is to identify phrases in the input data that are semantically and lexically similar, cluster them, and use the clusters to map phrases onto semantic slots. Algorithm 1 shows the detailed steps involved and Figure 1 presents an illustration. *First*, unlabelled input sentences (containing the new slots or concepts) are split into clauses[2] based on heuristics of punctuation and word connections. The output is a list of clauses–see Step 1 in Figure 1. *Second*, the clauses are grouped using unsupervised clustering as described in Section 3.1. *Third*, we map all clusters found onto the slot types and definitions in the system's ontology. New slots are identified from incoming user queries and extend the domain ontology. Their values do not matter for our algorithm, though. The mapping in this step is based on the similarity between the centroids of clusters and the definitions of slots. These definitions are based on phrases or keywords—see Step 3 in Figure 1. *Fourth*, the clustered clauses are chunked into phrases using a shallow parser that uses a combination of classifiers [22]. *Fifth*, all phrases found in Step 4 are classified as either representing the value of a slot or not. This is done using a Bayesian classifier, which was trained on non-lexical features of known slots, see Section 3.1.1. *Last*, the slot values of all slots identified in Step 5 are replaced by their corresponding semantic slot type–as derived from the cluster-slot mapping in the third step.

[2]A clause is the smallest grammatical unit that can express a complete proposition—next below the sentence in rank—and made up of phrases.

Algorithm 1 Automatic labeller of semantic slots

function SLOTLABELLER(List unlabelledSentences, Dictionary slots)
  clauses ← unique clauses in unlabelledSentences
  semanticMap ← {} ▷ cluster-slot mapping
  labelledClauses ← {} ▷ output
  affinityScores ← similarity between clauses (x_i, x_j), for $x_i \neq x_j$
  clusteredClauses ← Clustering(clauses, affinityScores, |slots|)
  for each cluster g in clusteredClauses **do**
    $centroid$ ← average similarity of clauses in cluster g
    $def(s_i)$ ← definition of slot s_i from the slots dictionary
    $s^* = \arg\max_{s_i \in slots}$ ClauseSimilarity(def(s_i), $centroid$),
    where def(s_i) is the definition of slot s_i in the slots dictionary
    semanticMap ← APPEND(g, s^*)
  end for
  for each clause c in clusteredClauses **do**
    $phrases$ ← Chunking(c)
    $phrase^* = \arg\max_{ph \in phrases} P(ph|\text{evidence}(ph))$
    $slotID^*$ ← semanticMap(cluster of clause c)
    $labelled$ ← clause c replacing $phrase^*$ by $slotID^*$
    labelledClauses ← APPEND($labelled$)
  end for
  return labelledClauses
end function

3.1. Clause grouping using unsupervised clustering

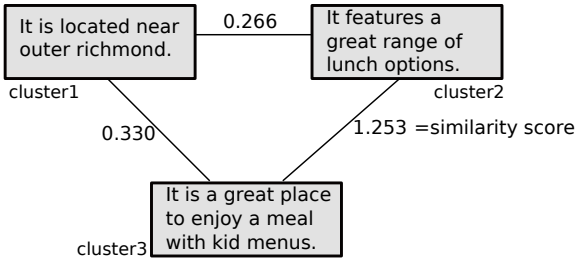
The task of our unsupervised clustering consists of partitioning clauses into maximally homogeneous groups, where homogeneity is measured based on numerical similarity. Here we describe a procedure for grouping clauses into k groups with equivalent semantics, corresponding to Step 3 in Algorithm 1. We assume that the number of clusters is known, e.g. in an interactive system the number of clusters may be defined by the number of new concepts (slots) raised by the user. We apply spectral clustering [23] due to its robustness to variant cluster shapes (other clustering methods are also possible).

In spectral clustering, given a set of data points $x_1, ..., x_n$ (clauses in our case) and a pairwise affinity matrix $A_{ij} = A(x_i, x_j)$, the task is to find a set of k clusters with a clustering algorithm of preference using the data points but projected into a low-dimensional space. Such a space is obtained according to the following procedure. First, construct the pairwise affinity matrix $A_{ij} = ClauseSimilarity(x_i, x_j)$,

Unlabelled Sentence (input): It is located near outer richmond, features a great range of lunch options, and is a great place to enjoy a meal with kid menus.

Step 1: Clause Splitting
It is located near outer richmond.
It features a great range of lunch options.
It is a great place to enjoy a meal with kid menus.

Step 2: Unsupervised Clause Clustering



Step 3: Cluster-Slot Mapping
cluster1 : def($nearlocation)=close to a landmark location
cluster2 : def($goodformeal) = good for meal, e.g. breakfast
cluster3 : def($kidsallowed) = allowed for children

Step 4: Clause Chunking
[It] [is located] [near] [outer richmond].
[It] [features] [a great range] [of] [lunch options].
[It] [is] [a great place] [to enjoy] [a meal] [with] [kid menus].

Step 5: Probabilistic Semantic Slot Identification
[It] [is located] [near] [outer richmond].
0.039 0.039 0.000 **0.992**

[It] [features] [a great range] [of] [lunch options].
0.034 0.119 0.039 0.000 **0.808**

[It] [is] [a great place] [to enjoy] [a meal] [with] [kid menus].
0.042 0.005 0.070 0.001 0.065 0.000 **0.817**

Step 6: Semantic Slot Labelling
[It] [is located] [near] $nearLocation.
[It] [features] [a great range] [of] $goodForMeal.
[It] [is] [a great place] [to enjoy] [a meal] [with] $kidsAllowed.

Labelled Clauses (output):
It is located near $nearLocation.
It features a great range of $goodForMeal.
It is a great place to enjoy a meal with $kidsAllowed.

Fig. 1. Example sentence in the restaurant domain showing the automatic labelling process of the proposed method. The input to this method is sentences in raw text, and the output is a set of clauses (simple sentences) annotated with semantic slots. Although Algorithm 1 assumes a set of sentences as input rather than only one sentence, this example shows only one input sentence for illustration purposes.

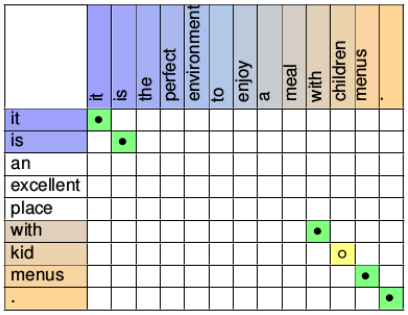


Fig. 2. Sample Meteor alignments with match markers: filled dots are exact matches, and the others are partial matches.

where the affinity between clauses x_i and x_j is defined by the following cumulative scores, each score in the range [0...1]:

$$ClauseSimilarity(x_i, x_j) = MS + WRA + SSS + STS, \quad (1)$$

explained as follows. MS (Meteor Score) measures the lexical similarity between sentences (see Figure 2) calculated as

$$MS = (1 - Pen) \times F_{mean}, \quad (2)$$

where F_{mean} is a weighted precision-recall metric and Pen is a penalty that accounts for gaps and differences in word order [24]. WRA (Word Recognition Accuracy) is the complement of the well-known 'Word Error Rate' metric and also measures the lexical similarity as

$$WRA = 1 - \frac{substitutions + deletions + insertions}{|words|}. \quad (3)$$

SSS (Semafor Semantic Score) measures the semantic similarity between feature vectors (see Figure 3) $F(x_i) = \{f_0^{x_i}, ..., f_N^{x_i}\}$ and $F(x_j) = \{f_0^{x_j}, ..., f_M^{x_j}\}$ of clauses x_i and x_j, which are extracted by the Semafor Frame-Semantic Parser [25]. The score is then expressed as

$$SSS = \frac{\sum_{m=1}^{|M|} \sum_{n=1}^{|N|} sim(f_n^{x_i}, f_m^{x_j})}{|F(x_i) \cap F(x_j)|}, \quad (4)$$

with function $sim(f_x, f_y)$ assigning 1 if semantic features $f_x = f_y$ and 0 otherwise.

Finally, STS (Semantic Textual Similarity) also measures the semantic affinity between word sequences x_i and x_j as

$$STS \approx sim_{LSA}(x_i, x_j) + 0.5 \exp^{-\alpha D(x_i, x_j)}, \quad (5)$$

where $sim_{LSA}(x_i, x_j)$ is the Latent Semantic Analysis (LSA) between clauses, $D(x_i, x_j)$ is the minimal path distance between terms within clauses derived from WordNet relations, and α is a weighting factor as described in [26]. The result of this first step is a matrix of affinity scores $A(x_i, x_j)$.

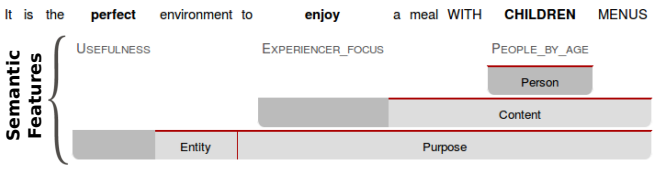


Fig. 3. Sample Semafor features within a clause.

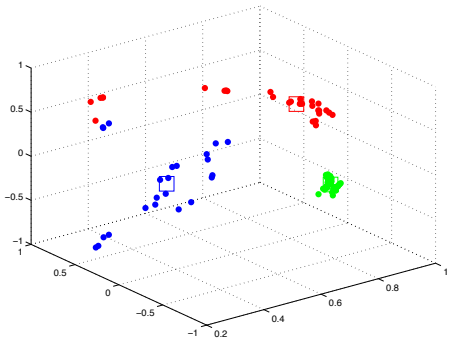


Fig. 4. Sample clustered clauses in 3D.

As a second step in spectral clustering, we compute the Laplacian matrix $L = D - A$, where D is the diagonal matrix with element (i, i) being the sum of row i in matrix A.

Third, we compute the first k eigenvectors $V = \{v_1, ..., v_k\}$ of the Laplacian matrix L. In linear algebra, an *eigenvector* v of matrix L satisfies the property $Lv = \lambda v$, where λ is a constant called *eigenvalue*. Let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of eigenvectors V.

Last, we cluster the data points y_i into k clusters. We used the K-means algorithm with the Manhattan distance defined by $d(p, q) = \sum_{i=1}^{n} |p_i - q_i|$, where p and q are eigenvectors. An example result of clustering a set of data points given the eigen-decomposition above, is illustrated in Figure 4 based on three slots assumed to be unknown (i.e., $k = 3$).

3.1.1. Semantic Slot Detection using Supervised Learning

Step 5 in Algorithm 1 identifies those phrases in the data that represent slots (in contrast to non-slot phrases). To do this, we use a Bayes net that was trained on features of known slots. The joint probability distribution for random variables (task-independent linguistic features in our case) Y is defined by $P(Y) = \prod_i P(Y_i|pa(Y_i))$, where $pa(.)$ denotes the set of parent random variables, and every variable is associated with a conditional probability distribution $P(Y_i|pa(Y_i))$. The following tasks are involved in the creation of our Bayes net: (1) structure learning involves constructing the dependencies of random variables based on the K2 algorithm [27]; and (2) parameter learning involves the estimation of conditional discrete probability distributions from data, where we use maximum likelihood estimation with smoothing. Once the Bayes net has been trained, we use the junction tree algorithm [28] for probabilistic inference, i.e., to compute the probabilities of phrases being semantic slots within a clause. The phrase with the highest probability is selected according to $\arg\max_{ph \in Phrases} P(ph|\mathbf{e}(ph))$, where the evidence of phrase ph is defined by $\mathbf{e}(ph) = \{f_i{=}val_i\}$ with features f_i.

Our Bayes net used the following features (binary except for the first two features): previous and next Part-Of-Speech (POS) tags [29], hasVerb, hasNoun, hasPronoun, hasAdverb, hasAdjective, hasPreposition, hasConjunction, phraseSize (small:$|words| \leq 4$, large:$|words|{>}4$), isStopWord, tf-idf level (low$\leq$5, high$>$5), and label (yes, no). The last feature was used to ask probabilistic queries, e.g. 'What is the probability of this phrase being a semantic slot?', and the remaining variables were used as evidence of the phrase at hand. An example probabilistic query to the trained Bayes net to determine how likely the phrase *"Union Square"* is to be a semantic slot is as follows: $Pr(label = yes|\mathbf{e}(\text{"}Union\ Square\text{"})) = 0.778$, where evidence **e** includes feature-values such as prevPOS=TO, nextPOS=END, hasNoun=true, phraseSize=large, isStopWord=false. These types of queries form the probability distribution of phrases (being semantic slots) in a clause. See Section 4.3 for classification accuracy in our domain.

4. EXPERIMENTS AND RESULTS

We describe two evaluations of our proposed method for automatic labelling of semantic slots, and its impact on a statistical surface realiser. First, we present an automatic evaluation to reveal the accuracy of clause clustering and semantic slot detection. Second, a user evaluation is presented to assess how humans perceive sentence generation for dialogue systems from automatically labelled data compared with manually labelled data.

4.1. The Data

Our domain is restaurants, where we assume a set of the semantic slots to be known (*venuename, food type, area*, and *pricerange*), and a set of slots to be unknown (*allowedforkids, goodformeal*, and *near* location). The latter are the ones that require automatic annotation. The corpus used for *unsupervised clustering* included 200 human-written sentences, each sentence containing 3 slots (which resulted in 600 clauses used for clustering). The unlabelled sentences served as input to Algorithm 1. In a second step, two independent human annotators labelled the data so that it could serve as a gold standard for our automatic labellings. The annotators discussed all diverging annotations and agreed on one. An

Metric	Accuracy (Purity)
Lexical Information (MS+WRA)	0.683
Semantic Information (SSS+STS)	0.842
Lexical+Semantic Information	**0.960**

Table 1. Experimental results in sentence clustering comparing lexical and semantic information, showing that its combination finds better clusters than individual metrics in isolation.

example sentence is provided in Figure 1. The corpus used for *supervised learning* is derived from 500 restaurant recommendations (from `www.list.co.uk`) containing the following slots: *venuename, foodtype, area*, and *pricerange*. For details on the corpus and annotations please see [30].

4.2. Results of Unsupervised Clause Clustering

In terms of unsupervised learning, we have applied the spectral clustering technique described in Section 3.1. It heavily relies on a set of numerical distances between clauses provided by four task-independent metrics: Meteor Score (MS), Word Recognition Accuracy (WRA), Semafor Semantic Score (SSS), and Semantic Textual Similarity (STS). These metrics were used because they provide affinities of lexical and semantic information. The motivation for using multiple metrics instead of a single metric is due to the lack of a task-independent metric providing meaningful distance scores between clauses. We compared the clustering accuracy (also referred to as 'purity') of lexical information (MS+WRA), semantic information (SSS+STS), and lexical plus semantic information (MS+WRA+SSS+STS), see Table 1. Purity is computed as $Purity(C,S) = \frac{1}{N}\sum_k max_j|c_k \cap s_j|$, where $C = \{c_k\}$ is the set of clusters, $S = \{s_j\}$ is the set of slots, and N is the number of clauses. While bad clusterings have purity values close to 0, good clusterings have purity values close to 1. It can be observed that the combination of lexical and semantic information achieves the best results. These results represent an improvement over the method proposed by [31], which is limited to only semantic information.

4.3. Results of Supervised Semantic Slot Detection

In terms of supervised learning, we trained a Bayes net (see Section 3.1.1) from restaurant recommendations (from www.list.co.uk) including known slots based on 2900 training instances. This data set was derived from labelled data of known slots (*venuename, foodtype, area, pricerange*). An evaluation on held-out data from known slots on a 10-fold cross validation reported a classification accuracy of 94.9%. To test the Bayes net's accuracy on unseen slots, we tested it on data containing our unknown slots (*allowedforkids, goodformeal*, and *near* location). This attained a precision P=0.672, recall R=0.955, F-measure F=2 $(\frac{P\times R}{P+R})$=0.789, and accuracy A=(true positives + true negatives)/all=0.912.

	Human	Labelled	Unlabelled
Understanding	**4.32**$^*\pm$0.92	4.06$\pm$0.88	4.03$\pm$0.91
Phrasing	**3.75**$^*\pm$1.19	3.13$\pm$1.20	3.03$\pm$1.27
Naturalness	**3.86**$^*\pm$1.09	3.41$\pm$1.08	3.33$\pm$1.18

Table 2. Average results from human ratings comparing utterances trained from manual and automatic slot labelling (no significant difference), contrasted with human written sentences (significant at p$<$0.003). *Significance based on a two-tailed Wilcoxon-Signed Rank Test, $\pm$ means std. deviation.

4.4. User Evaluation

We evaluated the output quality of a surface realiser trained from automatic slot labelling and compare it with a surface realiser trained from human annotations. As a first step, we use the output of our proposed method (a list of labelled clauses) for extending an existing corpus of labelled sentences [30], which resulted in 1300 sentences with different combinations of known slots (*venuename, foodtype, area, pricerange*) and unknown slots (*allowedforkids, goodformeal*, and *near* location). The former were manually labelled and the latter were automatically labelled. This extended corpus of labelled sentences was used to train a Conditional Random Field (CRF) based surface realiser for extended domains (i.e., for the known and unknown slots). In total the surface realiser can handle 7 slots, 4 referred to as 'known' and 3 referred to as 'unknown'. In addition, a second CRF trained only from human annotations (see Section 4.1 for the gold standard) was used as a baseline / upper-bound system. For details on the surface realiser, please see [30]. 50 sentences were generated for each surface realiser and rated by crowd-sourced users. An example sentence is *"This venue is called Jasmine Garden. It is close to the Duboce Triangle, and allows you to enjoy your meal with children."*

We ran a human rating study using crowdsourcing[3]. 202 users took part in our rating study and rated altogether 1908 utterances (containing repetitions) for their *understandability*, *phrasing* and *naturalness*. For understandability, the question we asked them was "Is this utterance understandable, i.e. is its meaning clear?"; for phrasing, the question was "Is this utterance well phrased?"; and for naturalness it was "Is this utterance natural, i.e. could it have been produced by a human?". Table 2 shows the mean ratings on 1-5 Likert scales, where 1 represents the worst and 5 the best. For comparison, the table also shows ratings for human-written sentences. Two results can be observed. First, none of our trained surface realisers was rated as well as the human-written sentences. Secondly, the outputs generated from automatic labels were rated very similarly to the outputs generated from human labels. There is no statistically significant difference between both sets of outputs according to a two-tailed Wilcoxon Signed-Rank test.

[3] `https://crowdflower.com/`

5. CONCLUSION AND FUTURE WORK

Previous work has treated statistical surface realisation mainly as a supervised learning problem, requiring labelled data. In this paper, we have addressed the problem of training a statistical surface realiser from unlabelled data, using automatic slot labelling. Our method uses unsupervised clustering to identify sentences with similar semantics based on a similarity function taking lexical and semantic features into account. We have applied spectral clustering due to its robustness to variant cluster shapes. In a second step, we have applied a Bayes net to distinguish phrases that represented semantic slots from those that do not. The automatically labelled data was used to train an existing surface realiser [30], which uses conditional random fields to generate outputs from input dialogue acts in the restaurant domain. An automatic evaluation showed over 90% of accuracy in identifying clusters and identifying slots. In addition, a human rating study compared the quality of utterances generated from automatically labelled data against utterances generated from human-labelled data. There was no significant difference between the two variants. This suggests that our automatic semantic slot labels were accurate enough to compete with human-labelled data.

Although we have extended this work with semi-supervised learning [32], in the future we aim to (1) evaluate the proposed method with additional unknown slots, data sets and algorithms; (2) compare the proposed clause similarity metric with other metrics to assess generalisation in growing domains and across domains; (3) automatically retrieve training data for our method from the web (rather than relying on non-automatic inputs), and (4) finally, perform an extrinsic evaluation with an end-to-end spoken dialogue system [33].

6. REFERENCES

[1] François Mairesse, Milica Gasic, Filip Jurcícek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young, "Phrase-based statistical language generation using graphical models and active learning," in *ACL*, 2010.

[2] Michael White, "CCG Chart Realization from Disjunctive Inputs," in *INLG*, 2006.

[3] S.R.K. Branavan, Harr Chen, Luke Zettlemoyer, and Regina Barzilay, "Reinforcement Learning for Mapping Instructions to Actions," in *ACL*, 2009.

[4] Adam Vogel and Dan Jurafsky, "Learning to Follow Navigation Directions," in *ACL*, 2012.

[5] Benjamin Snyder and Regina Barzilay, "Database-Text Alignment via Structured Multilabel Classication," in *IJCAI*, 2007.

[6] Regina Barzilay and Lillian Lee, "Bootstrapping Lexical Choice via Multiple-Sequence Alignment," in *EMNLP*, 2002.

[7] Gabor Angeli, Percy Liang, and Dan Klein, "A Simple Domain-Independent Probabilistic Approach to Generation," in *EMNLP*, 2010.

[8] Ioannis Konstas and Mirella Lapata, "Unsupervised Concept-to-Text Generation with Hypergraphs," in *NAACL*, 2012.

[9] Gökhan Tür, Asli Çelikyilmaz, and Dilek Hakkani-Tür, "Latent semantic modeling for slot filling in conversational understanding," in *ICASSP*, 2013.

[10] Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke Zettlemoyer, "A Generative Model for Parsing Natural Language to Meaning Representations," in *EMNLP*, 2008.

[11] Percy Liang, Michael Jordan, and Dan Klein, "Learning Semantic Correspondences with Less Supervision," in *ACL*, 2009.

[12] Pablo Duboue and Kathleen McKeown, "Statistical Acquisition of Content Selection Rules for Natural Language Generation," in *EMNLP*, 2007.

[13] Yuk Wah Wong and Raymond Mooney, "Generation by Inverting a Semantic Parser That Uses Statistical Machine Translation," in *NAACL/HLT*, 2007.

[14] Wei Lu, Hwee Tou Ng, and Wee Sun Lee, "Natural Language Generation with Tree Conditional Random Fields," in *EMNLP*, 2009.

[15] Nina Dethlefs and Heriberto Cuayáhuitl, "Comparing HMMs and Bayesian networks for surface realisation," in *NAACL HLT*, 2012.

[16] Matt MacMahon, Brian Stankiewicz, and Bejamin Kuipers, "Walk the Talk: Connecting Language Knowledge, and Action in Route Instructions," in *AAAI*, 2006.

[17] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy, "Toward Understanding in Natural Language Directions," in *HRI*, 2010.

[18] David Chen and Raymond Mooney, "Learning to Interpret Natural Language Navigation Instructions from Observations," in *AAAI*, 2011.

[19] Nina Dethlefs and Heriberto Cuayáhuitl, "Hierarchical Reinforcement Learning for Adaptive Text Generation," in *INLG*, 2010.

[20] Nina Dethlefs and Heriberto Cuayáhuitl, "Hierarchical Reinforcement Learning for Situated Natural Language Generation," *Natural Language Engineering*, vol. FirstView, august 2014.

[21] Srinivasan Janarthanam and Oliver Lemon, "Learning to Adapt to Unknown Users: Referring Expression Generation in Spoken Dialogue Systems," in *ACL*, 2010.

[22] Vasin Punyakanok and Dan Roth, "The use of classifiers in sequential inference," in *NIPS*. 2001, MIT Press.

[23] Ulrike von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, 2007.

[24] Michael Denkowski and Alon Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *EACL Workshop on Statistical Machine Translation*, 2014.

[25] Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith, "Probabilistic frame-semantic parsing," in *HLT*, 2010.

[26] Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and JonathanWeese, "UMBC EBIQUITY-CORE: Semantic textual similarity systems," in **SEM*, 2013.

[27] Gregory F. Cooper and Eduard Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, 1992.

[28] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[29] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *NAACL*, 2003.

[30] Nina Dethlefs, Helen Wright Hastie, Heriberto Cuayáhuitl, and Oliver Lemon, "Conditional random fields for responsive surface realisation using global features," in *ACL*, 2013.

[31] Yun-Nung Chen, William Yang Wang, and Alexander I. Rudnicky, "Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing," in *ASRU*, 2013.

[32] Heriberto Cuayáhuitl, Nina Dethlefs, and Helen Hastie, "A semi-supervised clustering approach for semantic slot labelling," in *ICMLA*, 2014.

[33] Helen Hastie and et al., "Demonstration of the PARLANCE system: a data-driven incremental, spoken dialogue system for interactive search," in *SIGDIAL*, 2013.