Supervised Clustering of Questions into Intents for Dialog System Applications

Iryna Haponchyk[1], **Antonio Uva**[1]*,
Seunghak Yu[2], **Olga Uryupina**[1] and **Alessandro Moschitti**[3,1]
[1]DISI, University of Trento, Povo (TN), Italy
[2]Samsung Research, Seoul, Korea
[3]Amazon, Manhattan Beach, CA, USA
iryna.haponchyk@unitn.it, antonio.uva@unitn.it
seunghak.yu@samsung.com, olga.uryupina@unitn.it, amosch@amazon.com

Abstract

Modern automated dialog systems require complex dialog managers able to deal with user intent triggered by high-level semantic questions. In this paper, we propose a model for automatically clustering questions into user intents to help the design tasks. Since questions are short texts, uncovering their semantics to group them together can be very challenging. We approach the problem by using powerful semantic classifiers from question duplicate/matching research along with a novel idea of supervised clustering methods based on structured output. We test our approach on two intent clustering corpora, showing an impressive improvement over previous methods for two languages/domains.

1 Introduction

The recent years have seen a resurgent interest for dialog systems, ranging from help desks and more complex task-based dialog to general purpose conversational agents, e.g., Alexa, Cortana or Siri. All these different application scenarios show that users expect to formulate complex information needs in natural language, with no limitation to their expressiveness. In other words, standard shallow semantic parsing based on concept segmentation and labeling is no longer sufficient for dialog modeling in today's applications. In particular, when designing dialog managers, we need to consider *user intents* expressed by articulated natural language text.

Current solutions to the intent detection problem consist in manually analyzing user questions and creating a taxonomy of intents to be attached to the appropriate actions. For example, if several semantically similar/identical questions regard `BookFlight`, the designer will build a category. This is a rather costly, difficult and time consuming task, which can often prevent the fast prototyping of dialog systems even for small domains. Moreover, the effort is extremely task- and system-specific, with very limited possibilities of porting the intent model, at least partially, across platforms, domains and languages. For example, in the recent 2016/2017 Natural Language Understanding Benchmarks (Coucke et al., 2017), the organizers have evaluated built-in intents generated by the major dialog managers (Apple's SiriKit, Amazon's Alexa, Microsoft's Luis, Google's Dialogflow, and Snips.ai) against a rather small set of relatively generic intents (e.g., `GetWeather`). This involved a considerable effort on aligning different system outputs.

To our knowledge, no previous work has been dedicated to automatizing this task, mainly because the underlying problem, semantic question paraphrasing, is very challenging. However, recent initiatives for automatic question duplicate detection[1], question relatedness (Nakov et al., 2016, 2017) and semantic textual similarity (Agirre et al., 2012; Cer et al., 2017) have shown that current technology achieves good accuracy in matching short text expressing similar semantics.

In this paper, we propose a model for automatically clustering questions into user intent categories, which can help the design of dialog systems. Our approach aims at overcoming the difficulty of providing a unique definition of intent from either a theoretical or practical perspective. Collaborating with stakeholders, we observe that it is very challenging to capture the intent property that can optimize dialogue/chatbot engineering work in terms of design and effort. The important aspect of our approach is that given a notion of intent, explicitly provided by annotated data, our

* The first two authors contributed equally to this manuscript.

[1]Quora: https://www.kaggle.com/c/quora-question-pairs

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2310–2321
Brussels, Belgium, October 31 - November 4, 2018.

model can create clusterings driven by such intrinsic definition. This is one of our major contributions: providing an effective supervised clustering approach, which can learn definitions from examples.

Regarding the technical solution our approach is bridging (i) the state-of-the-art methods for question similarity/paraphrasing with (ii) powerful supervised clustering algorithms. The former are obtained by exploiting previous research, e.g., on Quora, whereas the latter, are obtained by capitalizing on the structured output machine learning methods used for coreference resolution (CR), e.g., (Yu and Joachims, 2009; Fernandes et al., 2014). The main difference with CR consists in substituting mentions and their vector representation with the one of entire questions. It should be noted that structured output methods require a representation of edges connecting questions, which we obtain from question similarity research.

To train our model, we define a clustering corpus by automatically deriving question clusters from the Quora data, complementing the available question pair annotation with the transitive closure of the semantic matching property. Additionally, since Quora data is noisy, we manually annotate a question sample, defining an intent hierarchical taxonomy. To test the applicability of our methodology across languages and domains, we run evaluation experiments on another intent-based corpus, a collection of FAQs for an Italian online service.

We evaluate our approach comparing it to the classical k-means using (i) tf-idf model, (ii) our learned question similarity, and (iii) the similarity with spectral clustering. The last two methods can be considered another relatively novel contribution. Our results show that our new structured output method for question clustering: (i) highly outperforms all its competitors with respect to the standard clustering accuracy/purity as well as the measures defined in the CR domain, (ii) provides clustering accuracy of 80% with respect to the original Quora annotation and a still valuable accuracy of 65% with respect to the intent classes designed by an expert in Dialog modeling. This is a promising result as our designer can only provide one of the many interpretations of the intent taxonomy, which can be effectively applied.

Finally, a particular strength of our approach lies in identifying new intents (singleton clusters). These unseen intents are clearly problematic for dialogue management and cannot be covered by traditional approaches (e.g., our unsupervised clustering baselines show much lower performance on singleton clusters).

2 Related Work

Intent is a key concept for building dialog systems and is therefore a central research topic in the area. In particular, recent general-purpose dialog systems have to rely on extensive intent modeling to be able to correctly analyze a wide variety of user queries. This has led to a considerable amount of research on data-driven intent modeling.

In particular, Xu et al. (2013) represent query's intent as trees and employ a procedure for mapping an NL query into a tree-structured intent. The problem of this approach is that a new set of intent trees is required for new domains. Kim et al. (2016); Celikyilmaz et al. (2011) use semi-supervised approaches with large amounts of unlabeled data to improve the accuracy in mapping user queries into intents. However, they still require a small amount of labeled data in order to learn a given intent. Chen et al. (2016a) train a Convolutional Deep Neural Network to jointly learn the representations of human intents and associated utterances. Chen et al. (2016b) propose feature-enriched matrix factorization to model open domain intents. This leverages knowledge from Wikipedia and Freebase to acquire information from unexplored domains according to new users' requests. Unfortunately, it also requires external knowledge bases to induce concepts appearing within the intents.

Approaching the same problem from the opposite direction, several studies investigate algorithms for automatic question clustering. Wen et al. (2001) propose to cluster together queries that lead to the same group of web pages that are frequently selected by users. Jeon et al. (2005) use machine translation to estimate word translation probabilities and retrieve similar questions from question archives. Li et al. (2008) try to infer the intent of unlabeled queries according to the proximity with respect to the labeled queries in a click graph. Beitzel et al. (2007) propose to automatically classify web queries from logs into a set of topics by using a combination of different techniques, either supervised or unsupervised. The extracted topics are further used

for efficient web search. Deepak (2016) presents MiXKmeans, a variation of k-means algorithm, suited for clustering threads present on forums and Community Question Answering websites. However, most techniques use unsupervised clustering to group similar questions/queries, without modeling intents. In contrast, our study relies on supervised clustering to learn intent-based similarity.

Finally, our work is related to a large body of research on dialog acts (Stolcke et al., 2000; Kim et al., 2010; Chen et al., 2018): our low-level intent labels (Table 1) can be seen as very fine-grained dialog acts (Core and Allen, 1997; Bunt et al., 2010; Oraby et al., 2017).

However, our paper's objective is different as our goal is not to rigidly define intents and then exploit them to derive a semantic interpretation. We focus on two contributions: first, we aim at providing a tool to help implementing dialog managers such that the designer can more easily create categories from precomputed clusters. Note that having in mind hundreds of questions to create intent category from scratch is clearly an exigent task.

Second, our approach can dynamically cluster questions with the same semantics, without any concept annotation. Indeed, the important concepts will be learned from the training data, which constitutes a much simpler annotation task than the creation of ad hoc dialogue acts. In particular, this can help with domain-specific intents: the domain-level semantics will be learned from data with no need for advanced manual engineering.

3 Question Clustering Algorithms

In this paper, we explore techniques for clustering questions into user intent. To this end, an input set of questions Q undergoes splitting into subsets (clusters), $c_i = \{q_j^i\}_{j=1}^{N_i}$, where q_j^i is the j-th question in the cluster i of size N_i and $\bigsqcup_i c_i = Q$. Each c_i is assumed to contain questions with the same intent, i.e., to represent a distinct intent. Performing a clustering of a new question set Q in an unsupervised manner may generally be troublesome due to the lack of any information about the structure of Q and the target number of distinct intents in it. To overcome this issue, we learn a clustering function from data annotated with gold question clusters.

In this work, we pose the task as a supervised clustering problem following the formulation by Finley and Joachims (2005). Having the training examples of the form $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, where each input $\mathbf{x}_i$ is a set of elements of some nature and $\mathbf{y}_i$ – the corresponding gold standard clustering of such a set, the goal is to learn a predictor $h : X \rightarrow Y$, from the space of sets X to the space of clusterings Y. Supervised clustering has been shown particularly effective for the NLP task of coreference resolution (Yu and Joachims, 2009; Fernandes et al., 2014). These models learn to infer an optimal clustering $\mathbf{y}$ of an input set $\mathbf{x}$ in a structured way, i.e., as one output object optimizing a global scoring function $f : X \times Y \rightarrow \mathbb{R}$. This is different from local models, which aggregate multiple clustering decisions taken with respect to pairs of elements.

At the test time, the global models draw predictions by finding

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in Y}{argmax}\ f(\mathbf{x}, \mathbf{y}). \quad (1)$$

In the following, we give the necessary details of the original approach of Yu and Joachims (2009) and its adaptation for clustering questions.

3.1 Structured Output Clustering

To facilitate the inference of the optimal clustering in Equation 1, Yu and Joachims represent clustering variables $\mathbf{y}$ using graph structures. For an input $\mathbf{x}$, they construct a fully-connected undirected graph G, whose nodes are elements x_i of the input $\mathbf{x}$ and edges are all the pairwise links between them (x_i, x_j).[2] Any spanning forest $\mathbf{h}$ on G straightforwardly translates into a clustering $\mathbf{y}$. The nodes, aka elements represented by them, in each connected component (spanning tree) of $\mathbf{h}$ are considered to belong to the same cluster. The authors incorporate the spanning forest structures $\mathbf{h}$ as latent variables and decompose the feature representation of the input-output pair $(\mathbf{x}, \mathbf{y})$, which is extended with $\mathbf{h}$, over the edges of $\mathbf{h}$:

$$\Phi(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \sum_{(x_i, x_j) \in \mathbf{h}} \phi(x_i, x_j). \quad (2)$$

They employ Kruskal's spanning algorithm (Kruskal, 1956) to infer the optimal $\mathbf{h}$, and, respectively, $\mathbf{y}$.

A linear model $\mathbf{w}$ is trained using the Latent formulation of the Structural SVM learner (LSSVM),

[2]Note that we distinguish between boldface letters which we use to denote structural input/output variables, and normal font letters - for their rather elementary constituents, and simple variables.

to score the output clusterings according to the function $f(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}, \mathbf{h})$. In fact, $\mathbf{w}$ learns to score the edges since the structural feature vector decomposes over the edges. However, imposing a structure onto the output is supposed to produce a better $\mathbf{w}$, which we test in our experiments described in Section 5.

Alternatively, we employ the same structural model with another learning algorithm – latent structured perceptron (LSP) by Sun et al. (2009); Fernandes et al. (2014).

3.2 Pairwise question similarity classifier

Our intent clustering algorithm relies on the pairwise similarity between questions (edge score). To provide an accurate estimation of question-question similarity, we build upon an extensive state-of-the-art research in semantic similarity for short text, more specifically, on our previous work (Filice et al., 2016, 2017; Barrón-Cedeño et al., 2016; Da San Martino et al., 2016) solutions/features shown effective in the shared tasks by Nakov et al. (2016, 2017); Agirre et al. (2012, 2013).

In such work, the classifier is trained with SVMs, which learn a classification function $f : Q \times Q \rightarrow \{0, 1\}$ on duplicate vs. non-duplicate pairs of questions belonging to the question set Q. The classifier score is used to decide if two questions in the dataset q_i and q_j are duplicate or not. We represent question pairs as vectors of similarity features derived between two questions.

Feature Vectors are built for questions pairs, (q_i, q_j), using a set of *text similarity* features that capture the relations between two questions. More specifically, we compute 20 similarity features $sim(q_i, q_j)$ using word n-grams ($n = [1, \ldots, 4]$), after stopword removal, greedy string tiling (Wise, 1996), longest common subsequences (Allison and Dix, 1986), Jaccard coefficient (Jaccard, 1901), word containment (Lyon et al., 2001), and cosine similarity.

4 Building Intent clusters

In this study, we rely on two datasets, providing some important insights on question semantics. None of them, however, fully annotates intents explicitly. Below we describe our approach for converting these resources into intent corpora, relying on an automatic procedure followed by a manual post-annotation step. Our intent corpora as well as the larger raw question clusters collections are available to the research community.[3]

4.1 Quora corpus

The original Quora task requires detecting whether two questions are semantically duplicate or not. The associated dataset contains over $404,348$ pairs of questions, posted by users on Quora website, labeled as a duplicate pair or not. For example, *How do you start a bakery?* and *How can one start a bakery business?* are duplicate, while *What are natural numbers?* and *What is a least natural number?* are not. Note that the coders label pairs in isolation, only having access to one pair to be labeled at a time on Quora website (answer base). The pairs to be labeled are not selected randomly. To make the task more challenging, as well as more useful for practical applications, the organizers only offer pairs of questions that are somewhat semantically related:

(3) q_1: How does an automobile works?
q_2: How does automobile R&D work?
(4) q_1: Will I lose weight if I fast ?
q_2: Why am I losing weight so fast ?
q_3: How can I lose my weight fast ?

In (3), the lexical material is very similar, yet the questions are rather distinct, as reflected in the Quora annotation. They also express very different user intents: while q_1 is a generic curiosity question about automobiles, q_2 is a practical request for information on R&D in the automotive industry. Example (4) shows why Quora duplicate detection task is very challenging and requires a very good level of NL Understanding: while these three questions are very similar on the surface level, they all convey distinct semantics.

4.1.1 Question clusters from Quora

Differently from the original task, in this work, we are interested in automatically acquiring intents from large question repositories. Given this, we need a corpus that contains clusters of questions annotated by their underlying intent. As a first step in this direction, we approximate intent clusters with the clusters of similar questions from Quora. These can be obtained by exploiting the pairwise annotation and relying on the transitivity property: for each pair q_1, q_2 annotated as a duplicate, we assign q_1 and q_2 to the same cluster;

[3] https://ikernels-portal.disi.unitn.it/repository/intent-qa

negative pairs (non-duplicate question) do not impact the clustering in any way.

This procedure has obvious limitations by design: (i) it will not give us any intent labels, only the clusters and (ii) it will not provide any hierarchy of intents, or any general/large intent categories. Still, it provides data for a large number of user-generated intents that fully manual annotation initiatives (e.g., Natural Language Understanding Benchmarks) cannot guarantee.

4.1.2 Manually annotating intent clusters

The use of Quora dataset to derive intent raises several potential issues: (i) no consistency is enforced across labels, (ii) duplicate or very similar Quora *answers* potentially pollute the annotation for their corresponding questions, (iii) specific decisions may depend on availability and granularity of the underlying answers, and (iv) the annotation of popular questions might be very spurious since the users have no access to all the other related questions. The first issue has also been noted by the Quora competition[4] participants, who found the same pairs of questions appearing several times in the training set with different labels. Moreover, we found numerous cases where the annotation does not respect the transitivity property:

(5) q_1: What are, if any, the medical benefits of fasting?
q_2: What are the benefits of water fasting?
q_3: What are the health benefits of fasting?

Here, the three independent coders have produced inconsistent labeling: although q_2 and q_3 are explicitly labeled as non-duplicate, they are both considered duplicates of q_1.

The second issue arises when the answer base contains (near-)duplicate entries. For example, the following two very similar questions are considered non-duplicates since they lead to two distinct answers:

(6) q_1: Which is better - DC or Marvel?
q_2: DC VS Marvel: which do you like more? [non-duplicate]

Note that this typically happens for rather popular questions that are therefore important to be analyzed correctly, either manually or automatically.

The third issue is extremely important for areas only partially covered by the answer DB. For example, for the set of questions, *Why is Saltwater Taffy candy imported in LOCATION?*, most LOCATIONs are covered by a generic answer, and all the corresponding questions are judged duplicates. However, some specific LOCATIONs, e.g., Fiji, have a dedicated answer and thus the corresponding questions form singleton clusters.

Finally, the annotation coherence problem arises for very popular areas covering a lot of closely related questions. Thus, more than 100 questions cover different aspects of *Weight Loss*. Since the coders do not have any access to all the questions on the same topic, the individual decisions are not coordinated, which leads to rather arbitrary partitioning of the area into clusters:

(7) **Gold Quora Cluster 1:**
How can I lose weight ?
What is the easiest way to lose weight faster ?
How can you lose weight quickly ?
How do I lose 7kgs in 2 weeks ?
What a great diet to lose weight fast and not make you hungry or keep on measuring portions ?
.. many more
Gold Quora Cluster 2:
How can I lose 3 kg in one week?
Gold Quora Cluster 3:
What are the good diets for weight loss ?
What is the best diet plan for weight loss?

To overcome these issues, we manually re-annotated a portion of the original Quora dataset with intent-based clusters. Starting from automatically induced clusters described above, we first re-assess the partitioning, correcting eventual mistakes and then assign intent-based labels to our clusters. Our labels are hierarchical, thus allowing for a better flexibility when designing dialog managers: the dialog flow/actions can be defined in terms of more generic (e.g., `Advice`) or more specific (e.g., `Advice-WeightLoss-diet`) intents, depending on different implementation considerations (query frequency for specific intents, overall importance for the application, difficulty of processing inter alia). To stay inline with the recent intent-based dialogue research, we also annotate *slots*, where applicable: entities external to the intent, yet playing an essential role for the correct semantic interpretation of the question. Table 1 shows an example of the intent-based cluster annotation.

4.2 FAQ: Hype Intent corpus

Our second corpus, Hype, allows for a more direct evaluation of the intent clustering algorithm. The data come from a set of questions asked by

[4] https://www.kaggle.com/c/quora-question-pairs/discussion/30435

Intent	Slots	Questions
Recommend-TourismCuisine	streetfood, Delhi	What are the best street food in Delhi ?
Recommend-TourismCuisine	streetfood, Delhi	What is the best street food in Delhi ?
Recommend-TourismCuisine	streetfood, Delhi	What are the best street foods in delhi ?
Recommend-TourismRestaurant	streetfood, Delhi	What are the best street food places of delhi ?

Table 1: Manually annotated intent clusters for Quora

users to a conversational agent, collected and manually processed for constructing a FAQ section for Hype—an online service that offers a credit card, a bank account number and an ibanking app to its customers. Unlike Quora, the questions are explicitly assigned to clusters by human annotators, and these clusters correspond to intents by construction. However, they do not have any informative labels and there is therefore no associated hierarchy. While this corpus provides very valuable data for our study, its main disadvantage is a very limited number of questions. Some questions are reported below:

(8) q_0: Cos'é HYPE? (What is HYPE?)
q_1: Volevo dei chiarimente di cos'é la app hype (I'd like to have more information about the hype app)
q_2: mi puó spiegare cose'é la app hype (could you please explain me what the hype app is?)
q_3: informazione applicazione hype (information about hype app)

At the current stage of our research, we use the FAQ/Hype corpus directly, with no automatic or manual adjustments as we did for Quora. We plan to annotate the FAQ categories with explicit intent labels in future work.

5 Experiments

We describe our comparative experiments on our clustering models on two corpora in two different languages.

5.1 Setup

We used two different corpora, described in Section 4:

The Quora Intent corpus contains 270, 146 and 212 clusters in the training, development and test sets. The clusters contain different numbers of questions, ranging from singletons to groups of 100+ questions. The singletons are the dominant group in the Quora dataset. This is probably due to the inclusion of non-duplicate questions that appear in the original Quora dataset. Overall, there are 1,334 questions distributed in 628 clusters (an average of 2.12 questions/cluster).

The FAQ/Hype corpus contains no small-size (< 3) clusters by construction since smaller clusters are typically not selected as FAQ entries. The largest groups of clusters are those of size 8 and 9. Overall, the FAQ Intent corpus contains 147 questions spread in 28 intent-based clusters, which correspond to an average of 5.25 questions/cluster.

Models To perform supervised clustering, we use: (i) the original implementation of the Latent *SVM*^{struct} [5] – LSSVM, and (ii) our implementation of the LSP algorithm based on the same clustering inference on undirected graphs using Kruskal's spanning algorithm.

We compare these approaches to two unsupervised clustering baselines: (i) spectral clustering (Ng et al., 2001), for which we employ the implementation from the *smile*[6] library, and (ii) relational k-means (Szalkai, 2013). The former implementation takes a matrix of pairwise similarities between data points as input, whereas the latter approach is a generalization of k-means to an arbitrary matrix of pairwise distances. Thus, they can be run on the scores relative to the question pairs (q_i, q_j).

We provide two variants of such scores: first, we run both the methods on the scores obtained from a binary pairwise similarity classifier, described in Section 3.2. Second, we run the clustering baselines on the tf-idf scores computed for the question pairs. Note that our use of the scores from a trained pairwise classifier introduces some supervision in standard unsupervised clustering approaches, originating new hybrid methods.

Parametrization LSSVM and LSP require the tuning of a regularization parameter, C, and of a specific loss parameter, r (penalty for adding an incorrect edge), which we select on the dev. set. We pick up C from $\{1.0, 10.0, 100.0, 1000.0\}$, and the r values from $\{0.1, 0.5, 1.0\}$. K-means and spectral clustering algorithms require the indication of the number of clusters k. In all of our

[5] www.cs.cornell.edu/~cnyu/latentssvm/
[6] http://haifengl.github.io/smile/

Model	Clustering				Pairwise Classification			
	Precision	Recall	F1	$CEAF_e$	Precision	Recall	F1	Accuracy
LSSVM	**80.16**	77.81	**78.96**	**63.68**	**43.74**	32.00	36.96	88.41
LSP	66.06	**91.64**	76.78	51.50	20.36	**76.85**	32.19	65.62
SVM + spectral clustering	72.06	62.40	66.89	47.04	28.07	3.52	6.26	88.80
SVM + k-means	70.76	66.58	68.60	53.87	31.03	7.92	12.62	88.35
tfidf + spectral clustering	72.06	62.92	67.18	52.96	33.90	4.36	7.72	**88.94**
tfidf + k-means	69.19	65.01	67.04	50.94	29.95	5.33	9.04	88.62
SVM					26.25	72.23	**38.50**	75.50

Table 2: Supervised vs. unsupervised clustering models and pairwise classification baselines on the test set, where the gold labels are from the original Quora annotation. Note that pairwise classification does not provide a good estimation of clustering accuracy.

experiments, we use the gold standard k of each example (clustering) as parameter to run the baseline approaches. This corresponds to comparing with an upperbound of the baselines.

Measures We compare the output clustering $\hat{\mathbf{y}} = \bigsqcup_j \hat{c}_j$ to the ground truth $\mathbf{y}^* = \bigsqcup_i c_i$, where c_i, in our case, are either the clusters obtained with transitive closure from Quora annotation or the manually annotated categories (see Section 4.1).

For evaluation purposes, we assign each cluster $\hat{c}_j$ to the most frequent gold class (cluster), i.e., $argmax_i|c_i \cap \hat{c}_j|$, and compute the average precision over the clustering as:

$$Precision = \frac{1}{N}\sum_{j=1}^{\hat{k}} max_i|c_i \cap \hat{c}_j|, \qquad (9)$$

where N is the number of questions to be clustered, and $\hat{k}$ is the number of output clusters. This number is exactly the standard clustering purity by Zhao and Karypis (2002). Since the purity is known to favor the clustering outputs with the large number of clusters, we interchange the roles of output and gold clusters, which gives us the clustering

$$Recall = \frac{1}{N}\sum_{j=1}^{k} max_i|\hat{c}_i \cap c_j|, \qquad (10)$$

where k is the number of gold standard clusters. We then compute F1 from the above measures.

The defined majority-class based clustering measure allows assigning more than one cluster to the same gold cluster. The coreference resolution metric CEAF (Luo, 2005; Cai and Strube, 2010) solves this issue by finding one-to-one alignment between the clusters in the output and in the ground truth, based on which the final score is computed. We use $CEAF_e$, the variant with the entity-based similarity, as an alternative evaluation measure.[7] Note that, although we split the data into samples, all the clustering measures we use, the majority-based, defined by equations 9 and 10, and CEAF, are computed over the whole test sets (not by averaging scores separately for each sample).

We evaluate the results as well in terms of the classification scores relative to the correctness of the models in detecting the pairs of questions with the same/different intent. This enables the comparison against the pairwise classification approaches and an evaluation of their impact. We compute the Precision, Recall, and Accuracy of question pairs with the same intent.

5.2 Experiments on Quora

Original question label-based evaluation: first, we test the models on the clustering data from the Quora corpus, derived as described in Section 4.1. We train LSSVM, LSP, and the SVM classifier on the training part. The results of all the models on the test set are depicted in Table 2. In terms of clustering accuracy (the left half of the table), the LSSVM approach outperforms all the clustering baselines, improving about 10 points the highest baseline model, i.e., SVM + k-means both in terms of F1 and CEAF. LSP, in this setting, shows a slightly worse F1 than LSSVM, producing a model with high recall.

To study the impact of the pairwise classifier, we consider all the pairs of questions predicted by the clustering approaches as belonging to the same cluster as positive and the rest – as negative. All the clustering models show high accuracy, superior to that shown by the SVM classifier, but this is mainly due to the fact that there are much more negative question pairs. Interestingly, only LSSVM, though, among all the clus-

[7] We used the version 8 of the official coreference scorer conll.cemantix.org/2012/software.html

Model	Clustering				Pairwise Classification			
	Precision	Recall	F1	$CEAF_e$	Precision	Recall	F1	Accuracy
LSSVM	**84.92**	51.76	64.32	49.72	33.24	7.86	12.71	78.07
LSP	71.36	**89.45**	**79.38**	**59.99**	37.22	**83.46**	**51.48**	68.03
SVM + spectral clustering	62.31	43.22	51.04	33.04	35.11	2.95	5.44	79.17
SVM + k-means	68.84	47.24	56.03	48.62	42.71	6.48	11.25	**79.23**
tfidf + spectral clustering	65.83	45.23	53.62	35.46	38.18	3.62	6.62	**79.23**
tfidf + k-means	65.83	47.24	55.00	38.49	29.31	9.43	14.26	76.98
SVM					**40.35**	62.33	48.99	73.62

Table 3: Supervised vs. unsupervised clustering models and pairwise classification baselines on the test set, where the gold labels are provided by the intent-based manual annotation on a portion of the test set.

tering model, approaches the classifier in terms of classification F1. However, the cluster accuracy depends on many factors in addition to the pairwise classification accuracy.

Intent-based evaluation In Table 3, we present the results obtained on the portion of the test set which we manually annotated with the intent clusters. Here, we apply the same LSSVM, LSP, and the SVM classifier models trained in the experiments of the previous paragraph. However, we recompute all the four unsupervised clustering baselines supplying them with the new k – the number of gold intent-based annotated clusters.

In spite of being trained on data with different style of annotation (clusters automatically derived form Quora annotation), which is potentially rather noisy, LSSVM is able to recover the new intent categories better than the baseline approaches in terms of all the clustering metrics. The difference from the closest unsupervised clustering approach, which is the same as in the previous experiment, is now reduced in terms of CEAF. However, the information about the number of clusters in the ground truth critically impacts on the accuracy. The LSP model scored the best with respect to the new annotation. The lower classification accuracy of LSSVM with respect to the pairwise classifier is expected as the cluster number changed notably with the new annotation.

5.3 Evaluation on the FAQ dataset

Due to the limited size of the FAQ HYPE dataset, we split it into two parts, each of which forms one sample. We use the one containing 19 out of 28 clusters for training and the other with the remaining 9 clusters – for testing. The training sample comprises 97 questions, while the test sample – 50. The plots in Figure 1 illustrate the performance of LSSVM and LSP in terms of the clustering F1 in confront to the clustering baselines.

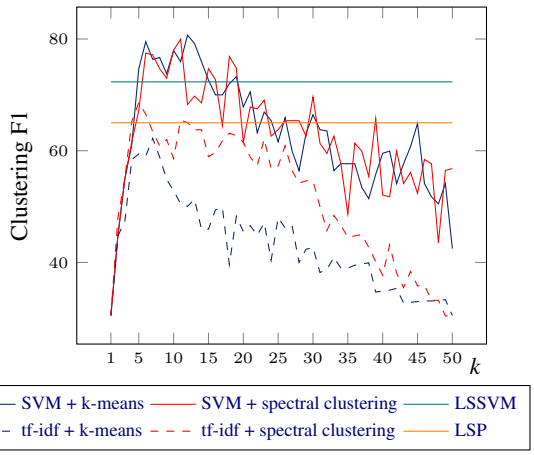

Figure 1: LSSVM and baseline clustering models; the latter vary with the cluster number k, on the FAQ HYPE test set.

We run the k-means and the spectral clustering algorithms with k in the range $(1, 50)$, which covers all the possible values for the given test set size.

LSSVM is better than the spectral clustering models with any k. k-means curves surpass LSSVM only in a narrow interval, showing high unstability. This suggests that guessing the k value in a realistic scenario in the absence of supervision does not seem an easy task. It should be also taken into consideration that we deal with very scarce training data. This also explains slightly insufficient accuracy of LSP compared to the k-means baseline.

5.4 Error Analysis and Discussion

As seen in tables 2 and 3, the structural output model consistently outperforms approaches based on spectral and k-means clustering on the Quora dataset. The most prominent improvement comes from singleton clusters: questions that are not duplicate with any other entries. Recall that the original dataset is constructed in such a way that singleton clusters are somewhat similar or related to existing material, but are still considered distinct by Quora annotators. LSSVM correctly recovers

71% of singleton clusters, whereas other methods perform much worse (5-30%). In the question-answering setting, singleton clusters correspond to novel questions that require setting up of a new entry in the answer base. Accurate recognition of singletons would allow for a timely allocation of resources to keep the answer base up-to-date and in line with incoming user requests.

Larger clusters are problematic for all the compared methods. Still, as evidenced by the CEAF score[8], the structural clustering is doing a better job at recovering non-singleton clusters. This mirrors our observations that even human annotators have difficulties correctly and consistently detecting duplicates in complex over-populated semantic areas (see Example 7) in the absence of the global context (e.g., list of all the related questions).

Finally, the clusters created by the LSSVM approach are more semantically related. Thus, 97% of all the suggested clusters contain questions with the same intent but, possibly, incorrect `slots`. For example, in the following question cluster:

(11) **gold cluster**
`Advice-Weightloss: fast,deadline`
q_1: How do I loose 50 lbs by Dec 2016?
q_2: How do I loose weight fast for operation ?
q_3: How can I lose 20 lbs super fast to audition for a small role in a movie ?
q_4: I want to lose weight for an event coming up in 2 weeks and I really don't care if I gain it back afterwords. What should I do ?

the user wants advice on losing their weight very fast by a specific deadline. LSSVM groups these questions with some others, more generic queries on fast weight loss (*How do I loose weight fast?*). This means that LSSVM captures the intent hierarchy well, providing meaningful clusters, although occasionally missing some important details. Other methods, on the contrary, form more poorly-related clusters (25-42% of clusters suggested by unsupervised approaches contain unrelated intents). Thus, the questions from Example (11) get grouped by other methods with such unrelated queries as *How is it to be in true love?* (spectral clustering over tf-idf).

Note that neither LSSVM nor unsupervised approaches have any access to the cluster labels and their hierarchy: in the training data, we only specify the clustering itself. Yet, by taking into account the global cluster structure, the LSSVM method can uncover the underlying hierarchy.

In the FAQ setting, most clusters are mid-size (5-9 questions). All the methods are doing a moderate job at recuperating the intent structure in this experiment. However, LSSVM shows better performance (cf. Section 5.3). Moreover, structural output is the only method that perfectly recuperates at least some clusters, e.g.:

(12) q_1: Non ricordo piú la password per accedere all'App (I don't remember the password for the App)
q_2: mi sono dimenticato la password (I forgot the password)
q_3: reimpostare la password (reset the password)
q_4: cambio password (change the password)

Here, LSSVM predicts the correct cluster exactly. K-means based approaches put $q_1 - q_4$ into the same cluster, however, they also merge them with *bloccare gli acquisti online (block the online purchases)*. Finally, spectral clustering does a poor job on this particular example, tearing either q_1 (tf-idf based spectral clustering) or q_2 (SVM pairwise classifier-based spectral clustering) apart and introducing a lot of spurious material.

6 Conclusion

We have proposed structured output methods fed with semantic question paraphrasing models to automatically extract user intents from question repositories. Our approach provides clustering accuracy of 80% with respect to the original Quora annotation and still valuable accuracy of 65% with respect one to of the many interpretations of question intent of our dataset, carried out by our expert in dialog modeling. This line of research looks promising as it can potentially simplify and speed up the work of Dialog Manager engineers. Although a deeper study is required to assess the benefits of our approach, the feedback of our designer clearly suggests that automatic clusters, even if were not perfect, simplify the annotation work. Several future research directions are enabled by our study, ranging from the use of neural clustering models to the application of our models to fast and semi-automatic prototyping of Dialog Systems. For this purpose, we make our data and software available to the research community.

Acknowledgements

This research has been supported by a grant from the Samsung Global Research Outreach Program: DiQuaVe Samsung GRO 2017.

[8]The reference scorer adopted by the coreference community discards singleton clusters.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43. Association for Computational Linguistics.

Lloyd Allison and Trevor Dix. 1986. A bit-string longest-common-subsequence algorithm. *Information Processing Letters*, 23(6):305–310.

Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq Joty, Alessandro Moschitti, Fahad A. Al Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. 2016. ConvKN at SemEval-2016 Task 3: Answer and question selection for question answering on Arabic and English fora. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, CA.

Steven M Beitzel, Eric C Jensen, David D Lewis, Abdur Chowdhury, and Ophir Frieder. 2007. Automatic classification of web queries using very large unlabeled query logs. *ACM Transactions on Information Systems (TOIS)*, 25(2):9.

Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. 2010. Towards an iso standard for dialogue act annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Jie Cai and Michael Strube. 2010. Evaluation metrics for end-to-end coreference resolution systems. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '10, pages 28–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

Asli Celikyilmaz, Dilek Hakkani-Tür, and Gokhan Tur. 2011. Leveraging web query logs to learn user intent via bayesian latent variable model.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14. Association for Computational Linguistics.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016a. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6045–6049. IEEE.

Yun-Nung Chen, Ming Sun, Alexander I Rudnicky, and Anatole Gershman. 2016b. Unsupervised user intent modeling by feature-enriched matrix factorization. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6150–6154. IEEE.

Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 225–234, New York, NY, USA. ACM.

Mark G. Core and James F. Allen. 1997. Coding dialogs with the damsl annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35, Cambridge, MA.

Alice Coucke, Adrien Ball, Clement Delpuech, Clement Doumouro, Sylvain Raybaud, Thibault Gisselbrecht, and Joseph Dureau. 2017. Benchmarking natural language understanding systems: Google, Facebook, Microsoft, Amazon, and Snips.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Salvatore Romeo, Antonio Uva, and Alessandro Moschitti. 2016. Learning to re-rank questions in community question answering using advanced features. In *Proceedings of the ACM Conference on Information and Knowledge Management*, CIKM '16, pages 1997–2000, Indianapolis, IN, USA.

P. Deepak. 2016. Mixkmeans: Clustering question-answer archives. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1585. Association for Computational Linguistics.

Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2014. Latent trees for coreference resolution. *Computational Linguistics*, 40(4):801–835.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2016. KeLP at SemEval-2016 Task 3: Learning semantic relations between questions and answers. In *Proc. of the 10th Intl. Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, USA.

Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2017. KeLP at SemEval-2017 task 3: Learning pairwise patterns in community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, pages 327–334, Vancouver, Canada.

Thomas Finley and Thorsten Joachims. 2005. Supervised clustering with support vector machines. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 217–224, New York, NY, USA. ACM.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 84–90, New York, NY, USA. ACM.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010. Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 862–871, Stroudsburg, PA, USA. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Scalable semi-supervised query classification using matrix sketching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 8–13.

Joseph Bernard Kruskal. 1956. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society, 7*.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2008. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Caroline Lyon, James Malcolm, and Bob Dickerson. 2001. Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 118–125, Pittsburgh, PA, USA.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48. Association for Computational Linguistics.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. Semeval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545. Association for Computational Linguistics.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 849–856, Cambridge, MA, USA. MIT Press.

Shereen Oraby, Pritam Gundecha, Jalal Mahmud, Mansurul Bhuiyan, and Rama Akkiraju. 2017. "how may i help you?": Modeling twitter customer serviceconversations using fine-grained dialogue acts. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces*, IUI '17, pages 343–355, New York, NY, USA. ACM.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373.

Xu Sun, Takuya Matsuzaki, Daisuke Okanohara, and Jun'ichi Tsujii. 2009. Latent variable perceptron algorithm for structured classification. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1236–1242, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Balázs Szalkai. 2013. An implementation of the relational k-means algorithm. *CoRR*, abs/1304.6899.

Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. 2001. Clustering user queries of a search engine. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 162–168, New York, NY, USA. ACM.

Michael J Wise. 1996. Yap3: Improved detection of similarities in computer program and other texts. In *ACM SIGCSE Bulletin*, volume 28, pages 130–134. ACM.

Juan Xu, Qi Zhang, and Xuanjing Huang. 2013. Understanding the semantic intent of natural language query. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 552–560. Asian Federation of Natural Language Processing.

Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1169–1176, New York, NY, USA. ACM.

Ying Zhao and George Karypis. 2002. Criterion functions for document clustering: Experiments and analysis. Technical report.