Intent Classification by the use of Automatically Generated Knowledge Graphs

Anonymous NAACL-HLT 2021 submission

Abstract

Intent classification is an essential task for goal-oriented dialogue systems, in order to automatically identify customers' goals. Although intent classification performs well in general settings, domain-specific user goals can still present a challenge for this task. To address this challenge, we automatically generate knowledge graphs for targeted datasets to capture domain-specific knowledge and leverage embeddings trained on these knowledge graphs for the intent classification task. We compare our results with state-of-the-art pre-trained sentence embeddings. Our evaluation on three datasets show improvement on the intent classification task in terms of precision.

1 Introduction

A large part of global business is in the consumer domain, providing services such as consumer payments, mobile cloud, and more. In providing these services to the customers, a business also needs to provide services to satisfy the customer needs that arise from their customer base (Temerak and El-Manstrly, 2019). Much of this support is provided through online interactions in the form of web chats. The ability to address these customer requests more efficiently can be of a significant business benefit.

The intent classification task is the automated categorisation of text, based on customer goals. It uses the concept of machine learning (ML) and natural language processing (NLP) to categorise a text string with different intents. In a general setting, a sentence like *"Where is the best place to buy a television?"* could be associated with the `purchase` intent. Since most goal-oriented dialogue systems are used to engage with customers through personalised conversations, intent classification is an essential component of these systems, where intent can be aligned with responses to a customer after typing in a question. The automated classification of user's intent reduces the manual effort for analysing user comments to identify avenues for improvements and issue remediation.

To enrich the classical classification task with domain-specific knowledge, we focus in this work on automatic Knowledge Graph (KG) generation. For this, we perform term extraction, named entity recognition (NER) and dependency parsing to align the concepts (terms and named entities) with semantic relations. We focus on publicly available datasets as well as on a proprietary domain-specific dataset in the telecommunication domain, where a classifier has to discriminate which utterance belongs to which intent class. For this, we generate automatically KGs based on the datasets used in this study. We distinguish between generic and domain-specific KGs. Since the automatically generated KGs are based on domain-specific data, they emphasise the depth of knowledge. We compare these results to a general KG, i.e., DBpedia (Lehmann et al., 2015), which is based on common knowledge and emphasises the breadth of knowledge. Within the process of KG generation, we evaluate the knowledge extraction, in particular, the extraction of entity classes and semantic relations between them, as expressed within the dataset. Finally, we leverage this information as Knowledge Graph Embeddings (KGEs) for intent classification according to extracted classes and relations.

2 Related Work

In this section, we provide an overview of related work focusing on intent classification using large pre-trained models and the incorporation of external knowledge for this task.

Zhang et al. (2019) demonstrate that informative entities in KGs can enhance language representation with external knowledge. The authors utilize large-scale textual corpora and KGs to train an enhanced language representation model, named ERNIE. The model can leverage lexical, syntactic, and knowledge information simultaneously. Zhang

et al. (2020) focus on the compositional aspects for intent classification. The authors decompose intents and queries into four factors (topic, predicate, object/condition, query type). To leverage the information they combine coarse-grained intents and fine-grained factor information applying multi-task learning. Purohit et al. (2015) study intent classification of short text from social media combining knowledge-guided patterns with syntactic features based on a bag of n-gram tokens. The authors explored knowledge sources (declarative, social behaviour about conversations and contrast patterns) to create pattern sets for examining improvement in the multiclass intent classification. The work demonstrated a statistically significant gain in performance on the dataset collected from Twitter only. By leveraging a knowledge-base and slot-filling joint model, He et al. (2021) propose a multitasking learning intent-detection system. The proposed approach has been used to share information and rich external utility between intent and slot modules. The LSTM and convolutional networks are combined with a knowledge base to improve the model's performance. Zhang et al. (2021a) proposed in their work IntentBERT, which is a pre-trained model for few-shot intent classification. The model is trained by fine-tuning BERT on a small set of publicly available labelled utterances. The authors demonstrate that using small task-relevant data for fine-tuning is far more effective and efficient than the current practice that fine-tune on a large labelled or unlabeled dialogue corpus. Siddique et al. (2021) propose an intent detection model, named RIDE, that leverages commonsense knowledge from ConceptNet in an unsupervised fashion to overcome the issue of training data scarcity. The model computes robust and generalisable relationship meta-features that capture deep semantic relationships between utterances and intent labels. These features are computed by considering how the concepts in an utterance are linked to those in an intent label via commonsense knowledge. Shabbir et al. (2021) present the generation of accurate intents for unstructured data in romanised Urdu and integrate this corpus in a RASA NLU module for intent classification. The authors embed the KG with the RASA framework to maintain the dialogue history for semantic-based natural language mechanism for chatbot communication and compare results with existing linguistic systems combined with semantic technologies. Hu et al. (2009) propose a general methodology to the problem of query intent classification by leveraging Wikipedia, one of the largest human knowledge bases. The Wikipedia concepts are used as the intent representation space, thus, each intent domain is represented as a set of Wikipedia articles and categories. The intent of any input query is identified through mapping the query into the Wikipedia representation space. The authors demonstrate the effectiveness of this method in three different applications, i.e., travel, job, and person name. Cavalin et al. (2020) explore intent classification where class labels are not represented as a discrete set of symbols but as a space where the word graphs associated with each class are mapped using typical graph embedding techniques. This allows the classification algorithm to take into account inter-class similarities provided by the repeated occurrence of some words in the training examples of the different classes. The classification is carried out by mapping text embeddings to the word graph embeddings of the classes. Their results demonstrate a considerable positive impact for the detection of out-of-scope examples when an appropriate sentence embedding such as LSTM and BERT is used. Ahmad et al. (2021) explored a joint intent classification and slot-filling task with unsupervised information extraction for KG construction. The authors trained the intent classifier in a supervised way but used this intent classifier for the slot-filling task in an unsupervised manner. They train a BERT based classifier for the intent classification task, which is used in a masking based occlusion algorithm, that extracts information for the slots from an utterance. A KG construction algorithm from dialogue data is also described in this paper. Within their evaluation, they observed that in a completely unsupervised setting the occlusion based slot-information extraction method yields good results. Furthermore, Pinhanez et al. (2021) leverage symbolic knowledge from curators of conversational systems to improve the accuracy of those systems. The authors use the context of a real-world practice of curators of conversational systems who often embed taxonomically-structured meta-knowledge, i.e. Knowledge Graphs, into their documentation. The work demonstrates that the Knowledge Graphs can be integrated into the dialogue system, to improve its accuracy and to enable tools to support curatorial tasks. Zhang et al. (2021b) focus on the performance of few-shot intent detection leveraging pre-training and fine-tuning approaches. Within the self-supervised con-

trastive pre-training approach the authors collected intent detection datasets without using any labels, where the model implicitly learns to separate fine-grained intents. In addition, the authors perform few-shot fine-tuning based on joint intent classification loss and supervised contrastive learning loss, where the supervised contrastive loss encourages the model to distinguish intents explicitly. Similarly, Liu et al. (2021) propose a new framework for few-shot intent classification and slot filling leveraging explicit-joint learning and supervised-contrastive learning. The authors demonstrate that explicit-joint learning utilises the close relationship between intent classification and slot-filling tasks, while supervised-contrastive learning benefits from more class-indicative representations.

Differently from the approaches mentioned above, our work focuses on providing domain-specific knowledge into the classification model, by automatically generating semantically structured resources, i.e. Knowledge Graphs, from the targeted datasets.

3 Experimental Setup

To observe the impact of the extracted information and the amount of extracted terms present in the KG on the intent classification task, we generated several KGs. We performed several NLP tasks, i.e., term extraction, taxonomy relation extraction, NER. We evaluated their performance separately by manually curating the automatically generated KGs on the proprietary ProductServiceQA dataset, which led to "Benchmark" KGs (cf. Table 2).

3.1 Knowledge Graph Creation

To automatically generate KGs from the targeted datasets, we used the KG extraction framework Saffron[1] (Bordea et al., 2013).

3.2 Knowledge Graph Embeddings

In a given KG, each subject h or object t entity can be associated as a point in a continuous vector space whereby its relation r can be modelled as displacement vectors ($h + r = t$) while preserving the inherent structure of the KG. In this work, we use TuckER (Balažević et al., 2019), a linear model based on Tucker decomposition of the binary tensor representation of KG triples. This allows us to create semantically-enriched KGEs that are used in the network embedding layers in our system.

[1]https://saffron.insight-centre.org/

3.3 Pre-trained Sentence-Embeddings

In this section, we provide a short description of these pre-trained models and how we used them to design our experiments. **LASER** (Artetxe and Schwenk, 2019) is a multilingual sentence encoder to calculate and use multilingual sentence embeddings. Created by Facebook Research, it learns joint multilingual sentence representations for 93 languages. It uses a single Bi-LSTM encoder combined with a decoder and is trained on publicly available corpora. LASER transforms sentences into language-independent vectors, which allows it to learn a classifier using training data in any of the covered languages. **SBERT** (Reimers and Gurevych, 2019) was designed to overcome the drawback of BERT or RoBERTa. While performing sentence-pair regression tasks, BERT or RoBERTa require that both the sentences should be fed into the network that leads to a massive computation overhead. SBERT uses a slightly different approach to construct semantically meaningful sentence embeddings. SBERT uses siamese and triplet network structures for generating the embeddings, which can be compared using cosine-similarity. **MPNet** (Song et al., 2020) is trained through permuted language modelling (PLM), allowing a better understanding of bidirectional contexts. In contrast to BERT, which neglects dependency among predicted tokens, MPNet leverages the dependency among predicted tokens through permuted language modelling and takes auxiliary position information as input to make the model see a full sentence and thus reducing the position discrepancy. The model is trained on various corpora (over 160GB of text) and fine-tuned on a variety of down-streaming tasks (GLUE, SQuAD, etc).

3.4 Datasets

The **ComQA** dataset[2] (Abujabal et al., 2018) consist of 11,214 questions of users' interest, which were collected from WikiAnswers,[3] a community question answering website. The dataset contains questions with various challenging phenomena such as the need for temporal reasoning, comparison, compositionality and unanswerable questions (e.g., *Who was the first human being on Mars?*). The questions in ComQA are originally grouped into 4,834 clusters, which are annotated with their answer(s) in the form of Wikipedia entities.

[2]http://qa.mpi-inf.mpg.de/comqa/
[3]https://www.answers.com/

	ProductServiceQA	ComQA	Paralex
# total samples	7,611	1,829	21,306
# samples (train)	4,795	1,097	12,784
# samples (val)	533	366	4,261
# samples (test)	2,283	366	4,261
# classes	338	272	275

Table 1: Statistics on the datasets used, i.e. ComQA, Paralex and ProductServiceQA dataset.

The **Paralex** dataset[4] (Fader et al., 2013) contains paraphrases, their word alignments, and basic NLP processed versions of the questions. There are about 2.5 million distinct questions and 18 million distinct paraphrase pairs. As an example, *"What are the green blobs in plant cells?"* and *a green substance in the plant cell be the ?* represent the question pairs within this dataset.

In addition to the openly accessible datasets, we further used a proprietary question-answer dataset, named **ProductServiceQA dataset**. It consists of 7,611 user queries, such as *"Can the VISA and MASTER cards be added to the card package?"*, which are distributed among 338 different classes (i.e. `Bank cards that can be added`).

To align the number of classes of all used datasets, we selected from ComQA only the QA pairs, which appear more than 6 times in the dataset. Similarly, to align a similar set to the ComQA and ProductServiceQA, we select the most frequent 275 classes from the Paralex dataset (Table 1).

4 Methodology

In this section, we provide insights on creating KGs from the targeted datasets, NER, dependency parsing for relation extraction and a relation filtering approach. Each step of KG generation allowed us to evaluate the impact of the semantic information represented in the KG. Table 2 illustrates the different KGs generated within this work. We conclude this section with the manual analysis of the automatically generated KGs.

4.1 Knowledge Graph Creation

The creation of domain-specific KGs follows a mixed approach based on the Saffron tool for taxonomy generation, novel NER approaches, relation extraction, triple filtering (Figure 1). Domain-specific terms and NEs are extracted from the corpus and used as a base for the generation of a taxonomy. Additional relations are extracted from the text corpus and filtered, before being added to the taxonomy to form a KG.

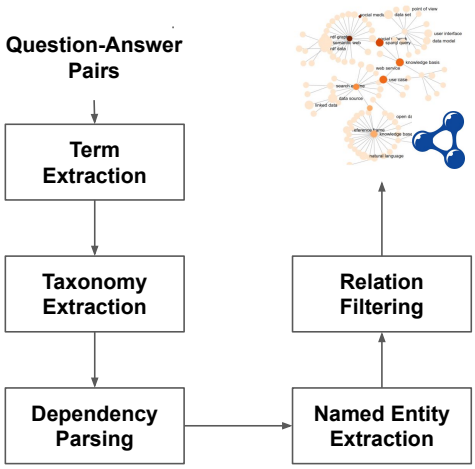

Figure 1: Knowledge Graph creation pipeline.

4.1.1 Taxonomy Generation

For taxonomy generation, we follow the approach by Pereira et al. (2019), where the **term extraction** module is a domain-independent approach, which is corpus-based and implements a four-step process: *(i)* identification of candidate terms, *(ii)* scoring, *(iii)* ranking, and *(iv)* filtering. The candidate term identification extracts noun phrases and uses other distribution metrics to select candidates. Then, a combination of scoring functions is used to measure the domain relevance of the terms (occurrence-based, context-based, using a reference corpus (e.g. Wikipedia), or based on topic modelling). Finally, terms are ranked by score and the top N is kept for the final list. The **taxonomy construction** step is constructing a taxonomy from the input set of terms extracted at the previous phase. For each distinct pair of concepts, c, $d \in C$, we attempt to estimate the probability, $p(c \sqsubseteq d)$. Based on the probability scores given by the Pairwise Scoring, a likelihood function is defined that represents how likely a given structure of concepts represents a taxonomy for the set of terms provided. Then, a search mechanism is used to find the taxonomy that maximizes the value of the likelihood function.

4.1.2 Named Entities Extraction

A domain-specific Named Entities (NEs) extraction model was built to extend the term extraction step to include NEs of relevance. A list of NEs that are specific to the dataset was provided and was used to train the NER system. Additionally, Flair was used to apply state-of-the-art NLP models.[5]

[4] http://knowitall.cs.washington.edu/paralex/

[5] https://github.com/flairNLP/flair

	Benchmark V1	Benchmark V2	Benchmark V3	Auto V1	Auto V2	Auto V3	Auto V4
Taxonomy	Y	Y	Y	Y	Y	Y	Y
Semantic Relations	N	Y	Y	N	Y	Y	Y
Named Entities	N	N	Y	N	N	Y	Y
Triple Filtering	N	N	N	N	N	N	Y
Unique Concepts	84	84	97	100	100	908	908
Unique Relations	1	221	221	1	230	259	157
Vocabulary	60	190	392	36	166	468	427

Table 2: Information on different KG information and statistics on the benchmarks and the automatically generated KG of the ProductServiceQA dataset.

Embedding	Prec.	Rec.	F1
Flair (Forward+Backward)	0.94	0.92	0.93
Flair (forward+backward) + GloVe	**0.95**	0.92	0.93
Flair (Forward)+GloVe	0.94	0.92	0.93
GloVe	0.92	0.91	0.91
BERT	0.93	0.91	0.93
ELMo	0.94	0.91	0.93

Table 3: Flair evaluation results for different embedding types.

		True Class	
		Positive	**Negative**
Predicted Class	**Positive**	97	16
	Negative	31	60

Table 4: Evaluation for the relation filtering model.

It provides multiple embedding methods, which can be used either individually or stacked to find the best fit for our dataset. After running several experiments with different combinations of stacked and individual embeddings, we have chosen `Flair(forward+backward)+GloVe` embedding as the best fit for our target domain. Table 3 gives the evaluation result of the experiments conducted on ProductServiceQA dataset.

4.1.3 Dependency-based Relation Extraction

This task makes use of dependency parsing to connect terms, based on a given corpus of texts. The corpus is parsed using the universal dependencies of the Stanford parser (Chen and Manning, 2014) implemented in the tool Stanza).[6] We replace *I*, *me* with *Customer* as the corpus contains questions from customers who refer to themselves. All dependencies involving a term (extracted previously using the Saffron framework) and a verb (using the POS information) are extracted. This provides a set of predicate-term pairs (`nsubj(pay, Customer)`, `obj(pay, bill)`). For phrasal verbs, particles are added to the predicate using an hyphen (-) (`get-up`), and for dependencies involving a preposition (*obl dependency type*), we concatenate the preposition to the predicate (`add_to, phone`). Triples (`term1, predicate, term2`) are constructed by combining any dependency pairs where, in the same sentence, the same predicate is the head of two dependencies in the list of pairs obtained in the previous step (e.g. `nsubj_obj(Customer, pay, bill)`). The triple relations are added to the existing Saffron-constructed taxonomy, by introducing a link labelled using the predicate as a relation between the two terms.

4.1.4 Relation Filtering

Relation filtering is a fully connected multi-layer perceptron model trained to identify a valid set of triples that are extracted from the dependency parsing step. The model is trained on both positive and negative sets of triples on the ProductServiceQA dataset. To obtain the negative set, we interchange subject and object and then evaluate existing triples for duplicates. If the negative triple is not present in the existing set, then we label this triple as a negative example. The evaluation for the relation filtering model is given in Table 4.

4.2 Sentence-Embedding Classification

We perform sentence embedding based intent classification that is built using some of the ideas presented in (Manjunath and McCrae, 2021). It is a multi-layer feed-forward neural network and the intuition behind it is that each dense layer learns a slightly more abstract representation. We create a sequential model. It is a fully connected network structure with five hidden layers. The dimension of the input layer is decided based on the dimension of the input embedding. The activation function used is `ReLU` (Nair and Hinton, 2010) and we use the *Sigmoid* function in the output layer. *Categorical Cross-Entropy* is used as loss function and *Adam* (Kingma and Ba, 2015) is used as the optimiser. We apply *Dropout* between the two hidden layers and between the last hidden layer and the output

[6] https://stanfordnlp.github.io/stanza/depparse.html

layer. We use 30% *Dropout* rate. The number of training epochs are 300 and batch size is 512.

The embeddings are fed through the above-explained network architrave for model building. KGEs are generated by running Tucker over the KG produced by Saffron. We also used other embedding such as GloVe, LASER, SBERT, and, MPNet in combination with KGEs. Basically, we generate an *n* dimensional embedding, where *n* varies based on the embedding method used. We use various sentence embedding techniques to perform the intent classification task. These various sentence embedding techniques can be categorised in three broad methods. In the first category, the network is trained with the state-of-the-art pretrained models, i.e., LASER, SBERT or MPNet. The results obtained from a single embedding category are considered baseline results. We performed **Concatenation** between LASER, SBERT, GloVe and KGEs. For a given sentence, two or more embeddings are concatenated to get the embedding matrix (E). A concatenation function is used to concatenate the different embedding vectors to get the final embedding vector. For **Substitution**, we are examining, if an embedding is present in the KG. If it is, we use KGEs otherwise GloVe embeddings. As both KG and GloVe have 300 dimensions the dimensions remain the same.

4.3 Manual Curation and Evaluation of KGs

We manually analysed and curated the automatically generated KGs, which yielded the "Benchmark" KGs that allowed us to evaluate the quality of the generated KGs. Three curators, one male and two female, all NLP specialists in term extraction, performed the curation.

Term Extraction Curation: The terms list was provided to the three annotators, where they independently identified terms that were correctly extracted, based on the definition of a term and the domain of the dataset. As an example, the extracted term `interconnection card free card` was annotated as incorrectly extracted term, while `interconnection card` was labeled as correct. Where possible, if the term span was incorrect, a corrected version was proposed. In this case, `wearable device support bank` was corrected to `wearable device`. The three annotators conferred to make a final decision. Within this manual curation step, 50% of terms were identified as correct , 13 terms were modified, and the Inter-Annotator Agreement (Fleiss Kappa) was 81%.

Taxonomic Relations Curation: A similar curation was performed on the extracted taxonomic relations . The curators were presented with pairs of terms involved in a taxonomic relation, i.e., parent_term → child_term, and had to identify whether the parent term was correctly identified for the child term (`flash payment` → `payment` - correct; `device` → `support` - incorrect). If the taxonomic relation was not correctly extracted, the experts proposed a replacement parent term from the list or a new term if none was deemed appropriate. Evaluating this step, 33% of relations were considered correct, with an Inter-Annotator-Agreement agreement of 70%. 20 new terms were defined and added to the taxonomy. This KG version contains 83 terms and the taxonomy has a depth of 5.

Named Entity with Dependency Relation Curation For the benchmark KGs, we collected a list of Named Entities (NEs) and their types, which resulted in 619 NEs (e.g. `card`) belonging to 22 different types (`CARD_TYPE`). In order to add the NEs to the KG, we selected the NE types that match a term in the taxonomy. Seven such types were identified. We then collected all the NEs corresponding to these seven types from the list (amounting to 25 NEs) and added them to their parent in the KG using a taxonomic relation.

The dependency-based relation extraction algorithm is performed, extracting predicates involving two NEs, or involving a NE and a term (from the initial list of terms in the third step of the approach (see 4.1.3). This list of triples with terms and NEs are finally added as relations that contain NEs to the previous KG. 126 new relations were added to the KG after curation, which showed 95% correctness and 79% Inter-Annotator agreement.

5 Results

Analysing the results for the **ComQA dataset**, MPNet embeddings contribute best to the classification task compared to LASER, SBERT or embeddings from the automatically generated KGs. Nevertheless, the performance of the KGs improves in relation to the number of terms within the KG. When concatenating sentence embeddings with GloVe or the automatically generated KGs, the `AutoV1` KG with 500 and 750 terms perform best (99.45), when they are combined with `LASER` and `SBERT` or `MPNET`. Comparing the performance between the GloVe embeddings and the automatically gen-

Method	Embeddings	Dim.	Precision
SOTA	SBERT	384	98.36
	LASER	1,024	96.75
	MPNet	768	98.63
	LASER+SBERT	1,408	98.28
	LASER+SBERT+GloVe	1,708	98.63
OURS	LASER+SBERT+AutoV1 (750)	1,708	**99.45**
	LASER+MPNet+AutoV1 (500)	2,092	**99.45**
	LASER+SBERT++AutoV1 (750)/GloVe	1,708	**99.45**

			AutoV1 (100)	AutoV1 (500)	AutoV1 (750)	AutoV2 (100)	AutoV2 (500)	AutoV2 (750)	DBpedia
	KG		40.71	75.41	86.89	45.08	75.13	83.61	14.92
Concat.	LASER+KG	1,324	95.35	95.62	95.08	95.63	95.08	95.08	96.17
	LASER+SBERT+KG	1,708	98.90	99.18	**99.45**	98.91	98.63	98.63	98.91
	LASER+MPNet+KG	2,092	99.18	**99.45**	98.09	98.91	98.36	98.63	98.36
Substit.	LASER+KG/GloVe	1,324	94.81	94.54	95.36	94.81	93.72	94.26	96.72
	LASER+SBERT+KG/GloVe	1,708	98.36	98.63	98.91	98.09	98.91	**99.45**	98.09
	LASER+MPNet+KG/GloVe	2,092	97.54	98.09	98.36	97.54	98.36	98.09	98.36

Table 5: Intent Classification evaluation for the ComQA dataset.

Method	Embeddings	Dim.	Precision
SOTA	SBERT	384	54.06
	LASER	1,024	52.92
	MPNet	768	53.80
	LASER+SBERT	1,408	54.07
	LASER+SBERT+GloVe	1,708	54.41
OURS	LASER+MPNet+KG	2,092	**55.40**

			AutoV1 (100)	AutoV1 (500)	AutoV1 (750)	AutoV2 (100)	AutoV2 (500)	AutoV2 (750)	DBpedia
	KG		22.38	46.67	49.39	25.86	47.82	47.65	20.15
Concat.	LASER+KG	1,324	54.04	54.39	54.72	53.94	54.74	54.48	53.24
	LASER+SBERT+KG	1,708	54.25	54.76	54.48	54.04	54.43	55.00	53.66
	LASER+MPNet+KG	2,092	54.48	**55.40**	54.81	53.89	55.07	55.16	53.66
Substit.	LASER+KG/GloVe	1,324	51.41	54.27	53.47	52.91	54.20	54.27	51.55
	LASER+SBERT+KG/GloVe	1,708	52.37	54.39	53.26	52.11	52.49	53.54	53.43
	LASER+MPNet+KG/GloVe	2,092	51.69	54.65	53.10	53.45	53.40	54.79	51.64

Table 6: Intent Classification evaluation for the Paralex dataset.

Method	Embeddings	Dimension	Precision
SOTA	SBERT	384	68.02
	LASER	1,024	62.68
	MPNet	768	69.25
	LASER+SBERT	1,408	68.60
	LASER+SBERT+GloVe	1,708	68.40
OURS	LASER+MPNet+KG	2,092	**70.00**

			Bench v1	Bench v2	Bench v3	Auto v1	Auto v2	Auto v3	Auto v4	DBpedia
	KG	300	26.19	34.91	38.10	25.62	31.80	45.15	39.33	23.61
Concat.	LASER+KG	1,324	63.20	62.06	62.46	63.64	63.16	63.42	63.03	62.77
	LASER+SBERT+KG	1,708	68.68	68.37	67.14	68.50	68.76	67.89	68.11	67.37
	LASER+MPNet+KG	2,092	68.77	68.94	68.24	69.51	68.16	68.77	69.21	**70.00**
Substit.	LASER+KG/GloVe	1,324	59.75	61.76	60.93	59.75	60.18	62.33	62.07	60.27
	LASER+SBERT+KG/GloVe	1,708	67.15	67.85	68.33	67.76	68.55	68.46	68.07	67.76
	LASER+MPNet+KG/GloVe	2,092	67.59	67.02	66.14	67.85	68.51	67.15	68.37	68.64

Table 7: Intent Classification evaluation for the ProductServiceQA dataset.

Method	Embeddings	Dimension	Precision
SOTA	SBERT	384	68.02
	LASER	1,024	62.68
	MPNet	768	69.25
	LASER+SBERT	1,408	69.39
	LASER+SBERT+GloVe	1,708	68.61
OURS	LASER+MPNet+KG	2,092	**69.99**

			Number of set Terms				
			100	200	300	500	1,000
	KG	300	40.34	40.34	41.61	42.14	44.20
Concatenation	LASER+KG	1,324	62.15	62.15	61.94	62.85	52.91
	LASER+SBERT+KG	1,708	68.24	68.24	67.89	67.85	67.85
	LASER+MPNet+KG	2,092	**69.99**	68.37	68.77	68.29	68.46
Substitution	LASER+KG/GloVe	1,324	62.51	60.58	61.54	62.64	60.36
	LASER+SBERT+KG/GloVe	1,708	68.20	68.37	68.20	67.81	67.41
	LASER+MPNet+KG/GloVe	2,092	67.89	67.90	67.19	67.76	67.24

Table 8: Impact of terms in the KG (AutoV3) for intent classification based on the ProductServiceQA dataset.

terms	100	200	300	500	1,000
Taxonomy			Y		
Semantic Relations			Y		
Named Entities			Y		
Triple Filtering			N		
Unique Concepts	908	1,008	1,108	1,308	1,808
Unique Relations	259	279	299	305	324
Vocabulary	468	494	529	553	653

Table 9: Statistics on the automatically generated KGs (AutoV3) with different thresholds of terms.

erated KGs, the latter outperforms the former in the majority of the setups. The substitution performs comparably to the concatenation approach, where combining LASER+SBERT+AutoV2 KG achieves the same precision as the best-reported concatenation approach.

For the **Paralex** dataset, leveraging SBERT pre-trained model performs best, when using it as a single resource (54.06). Although extracting more terms by the Saffron tool for KG creation improves the classification task, it does not reach the performance of the large pre-trained models. On the other hand, AutoV2 KG with 750 terms in combination with LASER+SBERT with performs best in the concatenation approach. In line with the previous experiments, the substitution approach demonstrates slightly worse results.

Furthermore, we leverage sentence embeddings on the proprietary **ProductServiceQA dataset** (Table 7). Analysing single embeddings, MPNet performs best (69.25), compared to SBERT, LASER or the automatically generated KGs and DBpedia. When combining sentence embeddings with the KGs, DBpedia contributes most in the concatenation approach with LASER+MPNet. Similarly to the results described above, embedding substitution does not outperform the concatenation approach.

At last, we analyse the impact of the set of terms within the KG, generated by the Saffron tool, which in its default setting will extract the 100 most domain-specific terms from the targeted document. Therefore, we extended this set gradually (Table 9). As seen in Table 8, extending the set of terms positively contributes when using the KGs as a single embedding resource. Nevertheless, even the KG with 1,000 terms does not outperform any pre-trained sentence embeddings used in this work. Nevertheless, when concatenating the KGs with these resources, LASER+MPNet+KG with 100 terms performs best.

6 Conclusion

In this work, we presented work on leveraging automatically generated knowledge graphs for intent classification. Along with the automatically generated Knowledge Graphs, we provide an analysis of each step towards their creation and provide insights on their evaluation and manual curation steps. We perform the intent classification using state-of-the-art sentence embeddings and combine these with domain-specific Knowledge Graph Embeddings, trained on the automatically generated Knowledge Graphs. We evaluate our methodology on three different datasets and demonstrate that the domain-specific knowledge within the semantically structured Knowledge Graphs further improves the intent classification task. Our ongoing work focuses on different neural architectures, such as Siamese networks, and the explainability of the classification outcomes.

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2018. Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. NAACL 2019.

Zishan Ahmad, Asif Ekbal, Shubhashis Sengupta, Anutosh Maitra, Roshni Ramnani, and Pushpak Bhattacharyya. 2021. Unsupervised approach for knowledge-graph creation from conversation: The use of intent supervision for slot filling. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Empirical Methods in Natural Language Processing*.

Georgeta Bordea, Paul Buitelaar, and Tamara Polajnar. 2013. Domain-independent term extraction through domain modelling. In *The 10th international conference on terminology and artificial intelligence (TIA 2013), Paris, France*.

Paulo Cavalin, Victor Henrique Alves Ribeiro, Ana Appel, and Claudio Pinhanez. 2020. Improving out-of-scope detection in intent classification by using embeddings of the word graph space of the classes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3952–3961, Online. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-Driven Learning for Open Question Answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Ting He, Xiaohong Xu, Yating Wu, Huazhen Wang, and Jian Chen. 2021. Multitask learning with knowledge base for joint intent detection and slot filling. *Applied Sciences*, 11(11):4887.

Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 471–480, New York, NY, USA. Association for Computing Machinery.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

Han Liu, Feng Zhang, Xiaotong Zhang, Siyang Zhao, and Xianchao Zhang. 2021. An explicit-joint and supervised-contrastive learning framework for few-shot intent classification and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1945–1955, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sampritha H Manjunath and John P McCrae. 2021. Encoder-attention-based automatic term recognition (ea-atr). In *3rd Conference on Language, Data and Knowledge (LDK 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*.

Bianca Pereira, Cecile Robin, Tobias Daudert, John P. McCrae, Pranab Mohanty, and Paul Buitelaar. 2019. Taxonomy extraction for customer service knowledge base construction. In *Semantic Systems. The Power of AI and Knowledge Graphs*, pages 175–190, Cham. Springer International Publishing.

Claudio Santos Pinhanez, Heloisa Candello, Paulo Cavalin, Mauro Carlos Pichiliani, Ana Paula Appel, Victor Henrique Alves Ribeiro, Julio Nogima, Maira de Bayser, Melina Guerra, Henrique Ferreira, and Gabriel Malfatti. 2021. *Integrating Machine Learning Data with Symbolic Knowledge from Collaboration Practices of Curators to Improve Conversational Systems*. Association for Computing Machinery, New York, NY, USA.

Hemant Purohit, Guozhu Dong, Valerie Shalin, Krishnaprasad Thirunarayan, and Amit Sheth. 2015. Intent classification of short-text on social media. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 222–228.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Johar Shabbir, Muhammad Umair Arshad, and Waseem Shahzad. 2021. Nubot: Embedded knowl-

edge graph with rasa framework for generating semantic intents responses in roman urdu. *ArXiv*, abs/2102.10410.

A. B. Siddique, Fuad T. Jamour, Luxun Xu, and Vagelis Hristidis. 2021. Generalized zero-shot intent detection via commonsense knowledge. *CoRR*, abs/2102.02925.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Mohamed Sobhy Temerak and Dahlia El-Manstrly. 2019. The influence of goal attainment and switching costs on customers' staying intentions. *Journal of Retailing and Consumer Services*, 51:51–61.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y.S. Lam. 2021a. Effectiveness of pre-training for few-shot intent classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1114–1120, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jian-Guo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021b. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1906–1912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jinghan Zhang, Yuxiao Ye, Yue Zhang, Likun Qiu, Bin Fu, Yang Li, Zhenglu Yang, and Jian Sun. 2020. Multi-point semantic representation for intent classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9531–9538.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Appendix

Question	Gold Standard	Single KG			Concat SBERT+KG
		DBpedia	Auto v1 (100)	Auto v1 (750)	Auto v1 (750)
which mountain range separates russia from georgia?	.../caucasus_ mountains	.../austria	.../caucasus_mountains	.../caucasus_mountains	.../caucasus_mountains
what states does west virginia border?	.../ohio	.../mexico_ city	.../red_sea	.../prague	.../prague
who is the first woman chief minister in india?	.../sucheta_ kriplani	.../mexico_ city	.../tricia_nixon_ cox	.../missouri	.../arkansas
what is the first book of alex rider?	.../storm-breaker	.../albert_a._ michelson	.../frost/nixon_(film)	.../monaco	.../united_arab_emirates
which country is right next to switzerland?	.../austria	.../maryland	.../the_curious_ case_of_benjamin_ button_(film)	.../northern_ireland	.../cathy_burge

Table 10: ComQA Intent Examples

Question	Gold Standard	Single KG			Concat LASER+SBERT	Substitution LASER+MPNet+KG/GloVe
		DBpedia	Auto v1 (100)	Auto v1 (750)		Auto v2 (100)
how many ounce in 1 litre bottle ?	how many ounce be a liter ?	1.75 liter ounce ?	1.75 liter ounce ?	how many calorie do a hamburger have ?	how many ounce be a liter ?	how many ounce be in one liter ?
what be the two zone that a glacier be divide into ?	what be two type of glacier ?	what be inappropriate subject matter for wikianswer ?	what be two type of glacier ?	what language do guyana speak ?	what be two type of glacier ?	what be two type of glacier ?
salary and job availability for a cardiologist ?	what be the yearly salary of a cardiologist ?	what be the yearly salary of nurse ?	how much do a esthetician make ?	how much do dental assistant get pay ?	what be the yearly salary of a cardiologist ?	what be the yearly salary of a cardiologist ?
how much be hamster in jollye pet shop new-townabby ?	how much do hamster cost in kearney ?	how much will a hamster cost with everything ?	how much will a pet hamster cost to by ?	what job can you get with a associte degree in education ?	how much will a hamster cost with everything ?	how much do hamster cost at pet co ?
does saturn have satellite if so how many ?	how many moon do saturn have ?	how many satellite do saturn have ?	how many satellite do saturn have ?	what natural resource do new jersey have ?	how many moon do saturn have ?	how many satellite do saturn have ?

Table 11: Paralex Intent Examples

Question	Gold Standard	Single KG			Concat LASER+SBERT+KG
		DBpedia	Auto v1 (100)	Auto v3 (750)	Auto v2 (100)
Which phones can use the Pay code?	Which of the following mobile phones support the Pay code?	How do I delete a bank card, traffic card, Eid, or door key if the phone is not repaired or sold or replaced by a non customer service center?	Say Hi	Pay is not displayed on the third-party app.	Which of the following mobile phones support the Pay code?
Failed to recharge the mobile phone during the full reduction activity. Solve the problem quickly.	Why does the phone number fail to be recharged all the time when the mobile phone is fully deleted?	How Do I Cancel the Automatic Renewal Service?	Which models support mobile phone recharge and full subtraction?	How Do I Participate in a Mobile Phone Recharge Amount Deletion Activity?	Why does the phone number fail to be recharged all the time when the mobile phone is fully deleted?
City Traffic Card opening fee adjusted to 16 cent.	City Traffic Card opening fee adjusted to 16 cent.	Traffic card opening service fee and card deletion and refund description	How Do I Cancel the Automatic Renewal Service?	City Traffic Card opening fee adjusted to 16 cent.	City Traffic Card opening fee adjusted to 16 cent.
The traffic card cannot be added.	Add a traffic card to the Pay.	The entrance for adding a traffic card to the Pay is not displayed.	Which cities can a traffic card be used in?	Handling Method of Traffic Card Recharge Failure	Failed to add a traffic card to the Pay.
What the hell is real name authentication?	What is real-name authentication?	How Do I Cancel the Automatic Renewal Service?	Pay method of deregistering real-name authentication (non-personal authentication)	Pay method of deregistering real-name authentication (non-personal authentication)	What is real-name authentication?

Table 12: ProductServiceQA Intent Examples