Title	Intent classification by the use of automatically generated knowledge graphs
Author(s)	Arcan, Mihael; Manjunath, Sampritha; Robin, Cécile; Verma, Ghanshyam; Pillai, Devishree; Sarkar, Simon; Dutta, Sourav; Assem, Haytham; McCrae, John P.; Buitelaar, Paul
Publication Date	2023-05-12
Publication Information	Arcan, Mihael, Manjunath, Sampritha, Robin, Cécile, Verma, Ghanshyam, Pillai, Devishree, Sarkar, Simon, Dutta, Sourav, Assem, Haytham, McCrae, John P., Buitelaar, Paul. (2023). Intent Classification by the Use of Automatically Generated Knowledge Graphs. Information, 14(5), 288. https://doi.org/10.3390/info14050288
Publisher	MDPI
Link to publisher's version	https://doi.org/10.3390/info14050288
Item record	http://hdl.handle.net/10379/17776
DOI	http://dx.doi.org/10.3390/info14050288

information

Article

Intent Classification by the Use of Automatically Generated Knowledge Graphs

Mihael Arcan [1,*], Sampritha Manjunath [1], Cécile Robin [1], Ghanshyam Verma [1], Devishree Pillai [1], Simon Sarkar [1], Sourav Dutta [2], Haytham Assem [3,†], John P. McCrae [1] and Paul Buitelaar [1]

[1] Insight SFI Research Centre for Data Analytics, Data Science Institute, University of Galway, Galway, H91 AEX4, Ireland
[2] Huawei Research, Dublin, D02 R156, Ireland
[3] Amazon Alexa AI, Cambridge, CB1 2GA, UK
* Correspondence: mihael.arcan@insight-centre.org
† This work was conducted when the author was working at Huawei Research, Ireland.

Abstract: Intent classification is an essential task for goal-oriented dialogue systems for automatically identifying 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 data sets to capture domain-specific knowledge and leverage embeddings trained on these knowledge graphs for the intent classification task. As existing knowledge graphs might not be suitable for a targeted domain of interest, our automatic generation of knowledge graphs can extract the semantic information of any domain, which can be incorporated within the classification process. We compare our results with state-of-the-art pre-trained sentence embeddings and our evaluation of three data sets shows improvement in the intent classification task in terms of precision.

Keywords: intent classification; term extraction; named entity extraction; relation extraction; Knowledge graph generation

Citation: Arcan, M.; Manjunath, S.; Robin, C.; Verma, G.; Pillai, D.; Sarkar, S.; Dutta, S.; Assem, H.; McCrae, J.P.; Buitelaar, P. Intent Classification by the Use of Automatically Generated Knowledge Graphs. *Information* **2023**, *1*, 0. https://doi.org/

Academic Editor: Ryutaro Ichise

Received: 30 March 2023
Revised: 28 April 2023
Accepted: 10 May 2023
Published:

1. Introduction

A large part of global business in the consumer domain is providing services, such as consumer payments, mobile cloud services, 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 [1]. Much of this customer support is provided through online interactions in the form of web chats. The ability to address these customer requests more efficiently can be of significant business benefit.

The intent classification task is the automated categorisation of text with different intents based on customer goals using machine learning (ML) and natural language processing (NLP) techniques. In a general setting, a sentence such as *"Where is the best place to buy a television?"* could be associated with the `purchase` intent. Because 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 a customer's asked question. Therefore, the automated classification of users' intent can significantly reduce the manual effort of 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, which is incorporated into the classification task. To automatically generate a KG, we leverage term extraction techniques, named entity recognition (NER), and dependency parsing to align the concepts, i.e., terms and named entities with semantic relations. We perform intent classification on two publicly available data sets, i.e., ComQA [2] and ParaLex [3], as well as on one proprietary domain-specific data set, named ProductServiceQA, in the telecommunications domain. For this, we automatically generate KGs based on the data sets used in this study, whereby we distinguish between generic and domain-specific KGs. Because 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 [4], which is based on common knowledge and emphasises the breadth of knowledge. Within the process of the automatic KG generation, we evaluate the knowledge extraction, in particular, the extraction of entity classes and semantic relations between them, as expressed within the data set. Finally, we leverage this information as knowledge graph embeddings (KGEs) for intent classification according to the 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 intent classification.

Leveraging large pre-trained embedding models for intent classification is explored in Cavalin et al. [5], 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 on the detection of out-of-scope examples when an appropriate sentence embedding such as LSTM and BERT is used. Similarly, Zhang et al. [6] proposed 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-tunes on a large labelled or unlabeled dialogue corpus. Furthermore, Zhang et al. [7] focused on the compositional aspects of intent classification. The authors decompose intents and queries into four factors, i.e., topic, predicate, object/condition, and query type. To leverage the information, they combine coarse-grained intents and fine-grained factor information applying multitask learning. Purohit et al. [8] studied the 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 to create pattern sets for examining improvement in the multiclass intent classification. The work demonstrated significant gains in performance on the data set collected from Twitter only.

Combining large pre-trained models with KGs is explored in Ahmad et al. [9], where the authors study 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 trained 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 yielded good results. Yu et al. [10] capture commonsense knowledge for e-commerce behaviours by semi-automatically constructing a KG for intent classification. The authors leverage large language models to semi-automatically construct an intention KG, which is then evaluated and curated by human annotators. The annotation is performed on a large number of assertions that can explain a purchasing or co-purchasing behaviour, whereby the intention can be an open reason or a predicate falling into one of 18 categories aligning with ConceptNet, e.g., `IsA`, `MadeOf`, `UsedFor`. Furthermore, Pinhanez et al. [11] manually leveraged 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. He et al. [12] presented their user intent system and demonstrated its effectiveness in downstream applications deployed in an industrial setting. For KG construction, the authors leveraged lexical rule matching, part-of-speech tagging, and short text matching to construct a KG with "isA" relations between the "intent" nodes.

Further work focused on leveraging large but generic knowledge bases or knowledge graphs for intent classification. Within this work, Zhang et al. [13] demonstrated that informative entities in KGs can enhance language representation with external knowledge. The authors utilized large-scale textual corpora and KGs to train an enhanced language representation model. The model can leverage lexical, syntactic, and knowledge information simultaneously. By leveraging a knowledge base and slot-filling joint model, He et al. [14] proposed a multitasking learning intent-detection system. The proposed approach was used to share information and rich external utility between intent and slot modules. The LSTM and convolutional networks were combined with a knowledge base to improve the model's performance. Siddique et al. [15] proposed 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 computed robust and generalisable relationship meta-features that capture deep semantic relationships between utterances and intent labels. These features were computed by considering how the concepts in an utterance are linked to those in an intent label via commonsense knowledge. Shabbir et al. [16] presented the generation of accurate intents for unstructured data in Romanised Urdu and integrated this corpus in a RASA NLU module for intent classification. The authors embedded the KG with the RASA framework to maintain the dialogue history for a semantic-based natural language mechanism for chatbot communication and compared results with existing linguistic systems combined with semantic technologies. Similarly, Sant'Anna et al. [17] engaged RASA to extract intents and entities from a given sentence. Using RASA, the authors investigated the effectiveness of automatic answering systems to consumer questions about products in e-commerce platforms. Hu et al. [18] proposed a general methodology for the problem of query intent classification by leveraging Wikipedia. The concepts in Wikipedia were used as the intent representation space, thus, each intent domain was represented as a set of Wikipedia articles and categories. The intent of any input query was identified by mapping the query into the Wikipedia representation space. The authors demonstrated the effectiveness of this method in three different applications, i.e., travel, job, and person name.

Differently from the approaches noted 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 data sets. This allows us to automatically generate a knowledge graph from a document of a targeted domain, which eliminates human intervention or the dependency on existing knowledge graphs needed to guide the intent classification within a goal-oriented dialogue system.

3. Experimental Setup

In this section, we provide information on the KG extraction framework, KGEs generation, the state-of-the-art (SOTA) pre-trained sentence embeddings, and the data sets used in this work.

3.1. Saffron—Knowledge Extraction Framework

To automatically generate KGs from the targeted data sets, we used the KG extraction framework Saffron (https://saffron.insight-centre.org/, accessed on 30. 03. 2023). The tool is designed to create a KG automatically from a large text corpus by identifying terms and relations between them using syntactic and corpus frequency information.

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. In this work, we use TuckER [19], which employs a three-way Tucker tensor decomposition, which computes the tensor T and a sequence of three matrices leveraging the embeddings of entities (A and C) and relations (B) between them ($G \approx T \otimes A \otimes B \otimes C$). This allows us to create KGEs that are used in the network embedding layers in our system.

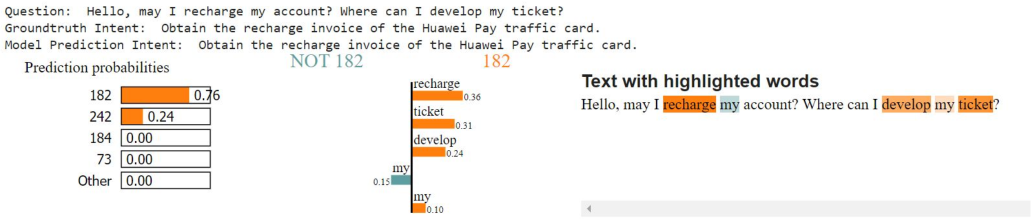
3.3. Pre-Trained Word and Sentence Embeddings

In addition to using the automatically generated KGs and KGEs trained on the targeted data sets, we leverage different SOTA pre-trained word and sentence embeddings for the classification task. First, we leveraged the pre-trained GloVe [20] word embeddings, which were trained on six billion tokens extracted from Wikipedia and the Gigaword archive (https://catalog.ldc.upenn.edu/LDC2011T07, accessed on 30. 03. 2023). LASER [21] is a multilingual sentence encoder to calculate and use multilingual sentence embeddings. The framework learns joint multilingual sentence representations for 93 languages and uses a single Bi-LSTM encoder combined with a decoder 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. Furthermore, we use SBERT [22], which uses Siamese and triplet network structures for generating sentence embeddings. Finally, we leverage MPNet [23], which is trained through permuted language modelling (PLM), allowing a better understanding of bidirectional contexts. MPNet leverages the dependency among predicted tokens through PLM and takes auxiliary position information as input to make the model see a full sentence. The model is trained on various corpora (over 160 GB of text) and fine-tuned on a variety of down-streaming tasks, such as GLUE and SQuAD, among others.

3.4. LIME

To better understand the predictions made by our intent classification models, we used the local interpretable model-agnostic explanations (LIME, https://github.com/marcotcr/lime, accessed on 30. 03. 2023.) algorithm introduced by Ribeiro et al. [24]. LIME learns an interpretable model locally around the prediction to explain predictions of any given classifier. For each prediction, it illustrates the degree how much each feature contributed to the outcome of the machine learning model's output. LIME implements this using the original model to generate fresh samples by making slight permutations to the feature values from the training set given to the model. Each of these samples is then given a weight based on its resemblance to the occurrence we are seeking to describe. The explainable model is then trained using the weighted proxy data created earlier.

Figure 1 illustrates the LIME visualisation of important words (i.e., `recharge`, `ticket`, `develop`) and their contribution degree for the intent classification of the question `Hello, may I recharge my account? Where can U develop my ticket?` Using LIME allowed us to identify the most important words that contribute to the classification task. Furthermore, we compare the top-k words (`k` = 5) provided by LIME with the automatically extracted terms using Saffron. This allowed us to filter the automatically generated KGs by Saffron based on the important words provided by LIME (cf. Section 4.3).



```
Question:  Hello, may I recharge my account? Where can I develop my ticket?
Groundtruth Intent:  Obtain the recharge invoice of the Huawei Pay traffic card.
Model Prediction Intent:  Obtain the recharge invoice of the Huawei Pay traffic card.
```

Figure 1. Visualisation of the LIME framework explanation with the top-k important words (k = 5) and their contribution degree based on the provided question.

3.5. Significance Testing

To compare the predictive accuracy of the two models, we use McNemar's test [25], which is based on a two-by-two contingency table of the two models' predictions. For McNemar's test, the null hypothesis shows there is no difference between the marginal frequencies. Therefore, if the *p*-value is greater than 0.05, it can be concluded that there is not a significant difference between false negatives and false positives. The alternative hypothesis shows there is a significant difference between the marginal frequencies, whereby the *p*-value is less than or equal to 0.05.

3.6. Data Sets

First, we used a proprietary question–answer data set, named ProductServiceQA data set (Table 1). 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`).

Table 1. Statistics on the data sets used, i.e., ComQA, ParaLex, ProductServiceQA, and ATIS.

	ProductServiceQA	**ComQA**	**ParaLex**	**ATIS**
# Total samples	7611	1829	21,306	5632
# Samples (train)	5328	1463	17,045	4833
# Samples (test)	2283	366	4261	799
# Classes	338	272	275	8

The ComQA data set [2] (http://qa.mpi-inf.mpg.de/comqa/, accessed on 30. 03. 2023) consists of 11,214 questions of users' interest, which were collected from WikiAnswers, a community question-answering website. The data set 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 4834 clusters, which are annotated with their answer(s) in the form of Wikipedia entities. To evaluate all data sets with a similar set of classes, we selected from ComQA only the QA pairs, which appear more than six times in the data set.

The ParaLex data set [3] (http://knowitall.cs.washington.edu/paralex/, accessed on 30. 03. 2023). The data set contains paraphrases, 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 data set. This allowed us to evaluate the targeted data sets with a similar set of intents, ranging between 272 intents for ComQA and 338 for ProductServiceQA.

The ATIS (Airline Travel Information Systems, https://www.kaggle.com/code/siddhadev/atis-dataset-from-ms-cntk?scriptVersionId=10371998, accessed on 30. 03. 2023) is a data set of manual transcripts regarding humans asking for flight information on automated airline travel inquiry systems. The data consist of 17 unique intent categories.

4. Methodology

In this section, we provide insights on using the automatically generated KGs from the targeted data sets, NER, dependency parsing for relation extraction, and a relation filtering approach. Each step of the KG generation allowed us to evaluate the impact of the semantic information represented in the KG in the classification task. As an example, Table 2 illustrates the different KGs generated from the ProductServiceQA data set. We conclude this section with the manual evaluation of the automatically generated KGs.

Table 2. Information on different KGs and statistics on the benchmarks and the automatically generated KGs of the ProductServiceQA data set.

	Benchmark KG$_t$	**Benchmark KG$_{tr}$**	**Benchmark KG$_{tre}$**	**KG$_t$**	**KG$_{tr}$**	**KG$_{tre}$**
Taxonomy	Y	Y	Y	Y	Y	Y
Semantic Relations	N	Y	Y	N	Y	Y
Named Entities	N	N	Y	N	N	Y
Unique Concepts	84	84	97	100	100	908
Unique Relations	1	221	221	1	230	259
Vocabulary	60	190	392	36	166	468

4.1. *Knowledge Graph Creation Pipeline*

The creation of domain-specific KGs includes NLP methods for term extraction, NER follows, and relation extraction provided by the Saffron tool for KG generation (Figure 2). To automatically generate KGs, 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 added to the taxonomy to form a KG.

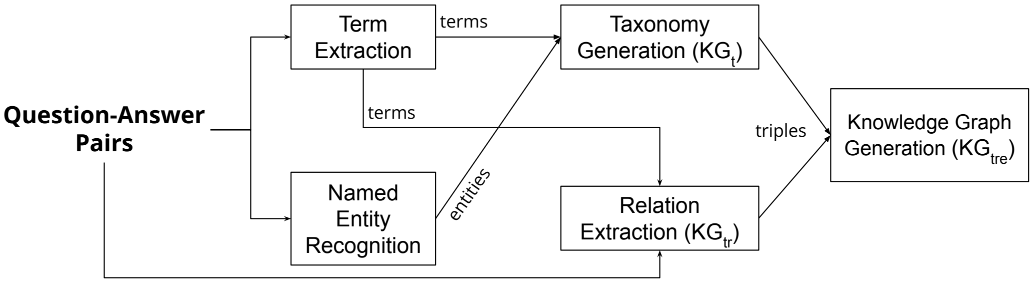

Question-Answer Pairs
Term Extraction
terms
Taxonomy Generation (KG$_t$)
terms
entities
Named Entity Recognition
Relation Extraction (KG$_{tr}$)
triples
Knowledge Graph Generation (KG$_{tre}$)

Figure 2. Knowledge graph generation pipeline.

4.1.1. Term Extraction

For the first step, we use the term extraction module implemented in Saffron [26]. The approach extracts noun phrases and uses distribution metrics to select term candidates. Then, a scoring function, i.e., occurrence frequency, context-relevance, reference corpus usage (e.g., Wikipedia), and topic modelling, is used to measure the domain relevance of the terms.

4.1.2. Named Entity Recognition

To obtain NEs from the targeted data sets, we used Flair's NER model, which is based on XLM-R embeddings (https://huggingface.co/flair/ner-english-large, accessed on 30. 03. 2023). To include domain-specific NEs of relevance for the proprietary ProductServiceQA data set, a domain-specific NEs recognition model was built to extend the term extraction step [27]. A list of NEs that are specific to the ProductServiceQA data set was provided and used to train the NER system. For this, we used Flair [28], more concretely the *"Flair* *(forward+backward)+GloVe"*, embeddings as they performed best for our targeted domain. Table 3 provides the results of a comparison of different embedding methods on the ProductServiceQA data set.

Table 3. Flair results for different embedding types on the ProductServiceQA data set.

Embedding	Precision	Recall	F_1
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

4.1.3. Taxonomy Generation

The taxonomy generation step is constructing a taxonomy using the top N ranked terms (Saffron's default setting is N = 100) and NEs obtained from the previous steps [29]. 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, greedy search is used to find the KG_t with a **t**axonomic IS-A relation that maximizes the value of the likelihood function.

4.1.4. Relation Extraction

To extract relations between the terms, we make use of dependency parsing. For this, the corpus is parsed using the universal dependencies of the Stanford parser [30] implemented in Stanza (https://stanfordnlp.github.io/stanza/depparse.html, accessed on 30. 03. 2023). All dependencies involving a term, extracted previously using the Saffron framework, and a verb (using the POS information) are extracted, which provides us with a set of predicate–term pairs, e.g., `nsubj(pay, customer)` or `obj(pay, bill)`. For phrasal verbs, particles are added to the predicate using a hyphen (-) (`get-up`). Similarly, for dependencies involving a preposition (*obl dependency type*), we concatenate the preposition to the predicate (`add_to, phone`). A triple (`term1, predicate, term2`) is 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 previously generated taxonomy (KG_t). The outcome of this step is the KG_{tr} with additional lexical **r**elations between the extracted terms.

4.1.5. Knowledge Graph Generation

By performing term extraction, NER, and relation extraction, we use the obtained triples to generate the KG_{tre}, incorporating **t**axonomic and lexical **r**elations between the extracted terms and named **e**ntities.

4.2. Intent Classification with Pre-Trained and Knowledge Graph Embeddings

Finally, we leverage the KGEs for intent classification trained on the KGs noted above using TuckER, which we combine with the pre-trained sentence embeddings. For this, we use a multi-layer feed-forward neural network. It is a fully connected network structure with five hidden layers, whereby the dimension of the input layer is decided based on the dimensions of the input embedding. The activation function used is `ReLU` [31], and we use the *Softmax* function in the output layer. *Categorical Cross-Entropy* is used as the loss function and *Adam* [32] is used as the optimiser. We apply *dropout* (0.3 dropout rate) between the two hidden layers and between the last hidden layer and the output layer. The number of training epochs is 300 and the batch size is 512.

The embeddings are fed through the above-explained network architecture for model building. With this, we leverage pre-trained embeddings (GloVe, LASER, SBERT, MPNet) in combination with KGEs. The various sentence embedding approaches used in our work can be categorised into three broad methods. In the first approach, the network is trained with the SOTA pre-trained models, i.e., LASER, SBERT, or MPNet. The results obtained from a single embedding category are considered our baseline results. Additionally, we performed a **Concatenation** approach, where, for a given sentence, two or more embeddings obtained from LASER, SBERT, GloVe, or KGEs are concatenated into the embedding matrix (E). For **Substitution**, we are examining if a term extracted from the data set is present in the KG. If it is, we use KGEs to obtain the embeddings; otherwise, GloVe embeddings are used. As both KG and GloVe have 300 dimensions, the input layer dimension remains the same.

4.3. Filtering Knowledge Graphs with LIME

We run the LIME algorithm on the targeted data sets and extract all words the model focuses on while making a prediction. The obtained ranked LIME list is compared with the top words provided by the Saffron tool. Whereas LIME extracts only unigrams, Saffron provides bi-grams extracted from the targeted data sets. We ran experiments with all KGs to obtain only the important words marked by LIME. This reduced the vocabulary of the KG by removing the excess noise.

4.4. Intent Classification on Intents Translated into English

Along with its intent in English, ProductServiceQA also holds the intents in Spanish and Chinese. To simulate the intent classification for these languages, we leveraged a translation pipeline to test how a slightly noisy data set affects the Siamese network classifier.

4.5. Manual Evaluation of KGs

We manually analysed and curated the automatically generated ProductServiceQA KGs, which resulted in the benchmark KGs for this data set. These benchmark KGs allowed us to evaluate the quality of the automatically generated ProductServiceQA KGs and are not used in training of the intent classification models or the generation of any other automatic KGs. Three curators, one male and two female, all NLP specialists in knowledge extraction, performed the curation.

Term Extraction Curation: The term list was provided to the three annotators who independently identified terms that were correctly extracted based on the definition of a term and the domain of the data set. As an example, the extracted term `pay card swiping` was annotated as an incorrectly extracted term, whereas `pay card` was labelled as correct. Where possible, if the term span was incorrect, a corrected version was proposed. In this case, `wearable device support bank` was corrected in the benchmark KGs to `wearable device`. Within this manual curation step, 50% of terms were identified as correct, whereby 13 terms were modified.

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 (hyponym) relation, i.e., parent_term → child_term. The annotators had to identify whether the parent term (`payment`) was correctly identified for the child term (`flash payment`). A wrongly identified relation pair would be `device` → `support`. 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 as correct, whereby 20 new terms were defined and added to the benchmark taxonomy. The benchmark KG$_{tr}$, which was used to evaluate the automatically generated KGs, contains 83 terms within a taxonomy of depth 5.

Named Entity with Dependency Relation Curation: For the benchmark KG$_{tre}$, we collected a list of 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 benchmark KG$_{tre}$, 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 benchmark KG$_{tre}$ using a taxonomic relation.

Within the same curation step, the dependency-based relation extraction algorithm was performed, extracting predicates involving two NEs, or a NE and a term (from the initial list of terms in the third step of the approach). A set of 126 triples with terms and NEs were finally added as relations that contain NEs to the previously mentioned benchmark KG$_{tre}$.

5. Results

In this section, we provide insights into experiments using an RNN as well as Siamese networks for the intent classification task. Additionally, we illustrate the performance of the classification task with the most important term in KGs, identified by the LIME framework. Finally, we evaluate the performance of the classification task in a multilingual setting.

5.1. Intent Classification with Recurrent Neural Networks

Analysing the results for the ComQA data set in the top part of Table 4, MPNet embeddings contribute best to the classification task compared to LASER or SBERT. The KGEs trained on the automatically generated KGs do not outperform the SOTA embeddings, although the performance of the KGs improves with the number of terms within the KG. As seen in Table A1 (see Appendix A), KG$_t$ with 100 terms achieves a precision of 40.71, whereas KG$_{tre}$, with 750 terms and relations between them, achieves a precision of 93.34.

When concatenating sentence embeddings with GloVe or the automatically generated KGs, `KG`$_t$ with 500 and 750 terms performs best (99.45) when it is combined with `LASER` and `SBERT` or `MPNET`. Comparing the performance between the GloVe embeddings and the automatically generated KGs, the latter outperforms the former in the majority of the setups. Substitution performs comparably to the concatenation approach, where combining `LASER+SBERT+KG`$_{tr}$ achieves the same precision as the best-reported concatenation approach.

For the ParaLex data set (Table 4), leveraging the SBERT pre-trained model as a single resource performs best (54.06). Nevertheless, when combining different embeddings, `LASER+SBERT+GloVe` outperforms the standalone embeddings (54.41). Similarly to the ComQA data set, although extracting more terms for KG generation improves the classification task (precision of 22.38 with KG$_t$ with 100 terms, 50.45 KG$_{tre}$ with 750 terms), it does not outperform any SOTA pre-trained models (Table A2 in Appendix A). On the other hand, in combination with `LASER+MPNet`, the KGEs trained on `KG`$_{tr}$ with 750 terms and extracted relations outperform the SOTA embeddings for the ParaLex data set using the `substitution` approach (55.42).

Next, we leverage the SOTA sentence embeddings on the proprietary ProductServiceQA data set (lower part of Table 4). Analysing single embeddings, compared to SBERT, LASER, or the KGEs trained on the automatically generated KGs and DBpedia, MPNet performs best on the proprietary data set in the telecommunication domain (69.25). When combining sentence embeddings with the KGs, DBpedia in combination with `LASER+MPNet` contributes the most when using the concatenation approach. Similarly to the data sets described above, embedding substitution does not outperform the concatenation approach (see Table A3 in Appendix A).

Table 4. Intent classification evaluation for the targeted data sets using an RNN (bold numbers indicate the best results for each setting).

ComQA Data Set			**ParaLex Data Set**		
SOTA Embeddings	Dimension	Precision	SOTA Embeddings	Dimension	Precision
SBERT	768	98.36	SBERT	768	54.06
LASER	1024	96.75	LASER	1024	52.92
MPNet	768	**98.63**	MPNet	768	53.80
LASER+SBERT	1792	98.28	LASER+SBERT	1792	54.07
LASER+SBERT+GloVe	2092	**98.63**	LASER+SBERT+GloVe	2092	**54.41**
Best Embeddings with KG	Dimension	Precision	Best Embeddings with KG	Dimension	Precision
LASER+SBERT+KG$_{t}$ (750)	2092	**99.45**	LASER+MPNet+KG$_{tr}$ (750)/GloVe	2092	**55.42**
LASER+MPNet+KG$_{t}$ (500)	2092	**99.45**			
LASER+SBERT+KG$_{tr}$ (750)/GloVe	2092	**99.45**			
ProductServiceQA Data Set			**ATIS Data Set**		
SOTA Embeddings	Dimension	Precision	SOTA Embeddings	Dimension	Precision
SBERT	768	68.02	SBERT	768	98.67
LASER	1024	62.68	LASER	1024	**98.87**
MPNet	768	**69.25**	MPNet	768	98.43
LASER+SBERT	1792	68.60	LASER+SBERT	1792	98.50
LASER+SBERT+GloVe	2092	68.40	LASER+SBERT+GloVe	2092	98.62
Best Embeddings with KG	Dimension	Precision	Best Embeddings with KG	Dimension	Precision
LASER+MPNet+KG (DBpedia)	2092	**70.00**	LASER+KG$_{t}$ (100)	1,324	**99.25**
			LASER+SBERT+KG$_{t}$ (100)	2092	**99.25**
			LASER+MPNet+KG$_{tr}$ (100)	2092	**99.25**
			LASER+MPNet+KG$_{tr}$ (100)	2092	**99.25**
			LASER+SBERT+KG$_{tre}$ (100)/GloVe	2092	**99.25**
			LASER+MPNet+KG$_{tre}$ (100)/GloVe	2092	**99.25**

In addition to the experiments noted above on the proprietary ProductServiceQA data set, we analysed the impact of the set of terms within KG$_{tre}$ extracted by the Saffron tool. As Saffron in its default setting extracts the 100 most domain-specific terms from the targeted document, we extended the set of domain-specific terms gradually (Table 5). As seen in Table 6, extending the set of terms positively contributes to the classification precision when using the KGs as a single embedding resource. As a result, even the KG with 1000 terms does not outperform any pre-trained sentence embeddings used in this work. Nevertheless, when concatenating the KGs with SOTA pre-trained embeddings, LASER+MPNet+KG$_{tre}$ with 100 terms performs best (69.99).

For the ATIS data set (bottom part of Table 4) in the aviation travel inquiry domain, LASER performs best (98.87) as a single embedding resource, whereas combining LASER with SBERT or SBERT+GloVe does not improve the performance on the classification task. In contrast, many systems that leverage the information of the KGs outperform the classification task when only the LASER pre-trained embeddings are used (99.25).

Table 5. Statistics on the automatically generated KG$_{tre}$ with different thresholds of terms.

Terms	100	200	300	500	1000
Unique Concepts	908	1008	1108	1308	1808
Unique Relations	259	279	299	305	324
Vocabulary	468	494	529	553	653

Table 6. Impact of different sets of terms within the $\mathtt{KG}_{tre}$ for intent classification, based on ProductServiceQA (bold numbers indicate the best results for each setting).

	SOTA Embeddings	Dimension	Precision	Best Embeddings with KG		Dimension	Precision
	SBERT	768	68.02	LASER+MPNet+KG$_{tre}$ (100)		2092	**69.99**
	LASER	1024	62.68				
	MPNet	768	**69.25**				
	LASER+SBERT	1792	68.60				
	LASER+SBERT+GloVe	2092	68.40				
			Number of Set Terms				
	Embeddings with KG	Dimension	100	200	300	500	1000
	KG	300	40.34	40.34	41.61	42.14	44.20
Concat.	LASER+KG	1324	62.15	62.15	61.94	62.85	52.91
	LASER+SBERT+KG	2092	68.24	68.24	67.89	67.85	67.85
	LASER+MPNet+KG	2092	**69.99**	68.37	68.77	68.29	68.46
Substit.	LASER+KG/GloVe	1324	62.51	60.58	61.54	62.64	60.36
	LASER+SBERT+KG/GloVe	2092	68.20	68.37	68.20	67.81	67.41
	LASER+MPNet+KG/GloVe	2092	67.89	67.90	67.19	67.76	67.24

5.2. Siamese Network

In addition to the experiments using the RNN architecture, we employed a Siamese network for the classification task. The top part of Table 7 illustrates the results for the **ComQA** data set, where compared to `LASER`, as well as the concatenation of `SBERT` and `LASER` embeddings, `SBERT` embeddings contribute best (95.18) to the intent classification task. When concatenating sentence embeddings with KGEs trained on the automatically generated KGs, $\mathtt{KG}_{tre}$ with 100 terms combined with `SBERT` and `LASER` performs the same as `SBERT` pre-trained embeddings only (95.18).

Table 7. Intent classification evaluation for the targeted data sets using a Siamese network (bold numbers indicate the best results for each setting; * denote statistically significant, p = 0.05).

ComQA Data Set			**ParaLex Data Set**		
SOTA Embeddings	Dimension	Precision	SOTA Embeddings	Dimension	Precision
SBERT	768	**95.18**	SBERT	768	48.81
SBERT+LASER	1792	94.66	SBERT+LASER	1792	49.75
MPNET	768	94.37	MPNET	768	50.33
MPNET+LASER	1792	94.14	MPNET+LASER	1792	**50.47**
Best Embeddings with KG	Dimension	Precision	Best Embeddings with KG	Dimension	Precision
SBERT+LASER+KG$_{tre}$ (100)	2092	**95.18**	MPNET+KG$_t$ (100)	1,068	**52.29** *
ProductServiceQA Data Set			**ATIS Data Set**		
SOTA Embeddings	Dimension	Precision	SOTA Embeddings	Dimension	Precision
SBERT	768	**73.94**	SBERT	768	**99.37**
SBERT+LASER	1792	73.77	SBERT+LASER	1792	99.00
MPNET	768	73.55	MPNET	768	98.62
MPNET+LASER	1792	73.51	MPNET+LASER	1792	98.62
Best Embeddings with KG	Dimension	Precision	Best Embeddings with KG	Dimension	Precision
MPNet+LASER+KGt_r (100)	2092	**74.69**	MPNet+KG$_{tre}$ (100)	1068	**99.50**

For the **ParaLex** data set (top right part in Table 7), the `MPNet` pre-trained model as a single resource performs best within the classification task (50.33). Next, when combining different embeddings, `MPNet+LASER` slightly improves the performance on the classification task (50.47). We significantly ($p < 0.05$) outperform the performance of the classification task of `MPNet+LASER` when leveraging the KG$_t$ with 100 terms in combination with `MPNet`.

The lower part of Table 7 illustrates the intent classification task on the **ProductServiceQA** data set using the Siamese network. Analysing SOTA pre-trained embeddings, SBERT performs best (73.94); when leveraging the KGEs trained on automatically generated KGs, the combination of MPNet+LASER+KG$_t t$ with 100 terms further improves the performance of the classification task compared to the existing pre-trained models (74.69 vs. 73.94).

For the **ATIS** data set, SBERT demonstrates the best performance among the SOTA pre-trained models (lower part of Table 7). When leveraging the KGEs based on the automatically generated KGs, MPNet+KG$_{tre}$, further improves the performance of the intent classification task.

As the sentences in the ComQA data set appear frequently and are thus repetitive, we filtered the data set with sentences in a manner so that they appear only between two and five times in the data set. Compared to the entire ComQA data set (top part of Table 7), the classification precision drops due to the smaller set of sentences used to train the Siamese network (≈95 vs. ≈84). Table 8 demonstrates best performance by the concatenation of the MPNet+LASER embeddings (84.23), while concatenating sentence embeddings with the automatically generated KGs, i.e., SBERT+LASER+KG$_t$ with 100 terms, outperforms the usage of SOTA embeddings (84.87 vs. 84.23). Table A6 in Appendix A illustrates the extended analysis with different KGs and sets of terms and relations extracted by Saffron.

Table 8. Intent classification evaluation for the ComQA data set, filtered by questions with a frequency between two and five, using a Siamese network (bold numbers indicate the best results for each setting).

SOTA Embeddings	**Dimension**	**Precision**
SBERT	768	83.31
SBERT+LASER	1792	84.12
MPNET	768	83.25
MPNET+LASER	1792	**84.23**
Best Embeddings with KG	Dimension	Precision
SBERT+LASER+KG$_t$ (100)	2092	**84.87**

5.3. Filtering Knowledge Graphs Using LIME

Within the next steps, we analysed the significance of the automatically extracted terms and relations within the KGs. Therefore, we leveraged the LIME toolkit (cf. Section 3.4) to exclude terms in the original KGs that are not considered important by LIME.

5.3.1. Filtering for Intent Classification with Recurrent Neural Networks

For the **ComQA** data set (top part of Table 9), we observed two cases, i.e., KG$_{tr}$ with 100 terms and KG$_{tr}$ with 750 terms, where the classification performance significantly ($p < 0.05$) improved compared to the original KGs (90.49 vs. 90.67 and 94.96 vs. 95.25). This demonstrates that these filtered KGs, which are reduced to one-third (or less) of their original size, hold important domain-specific information to guide the classifier to predict the correct intent.

For the **ParaLex** data set, three filtered KGs significantly improve ($p < 0.05$) the performance of the classification task over the original KGs (54.72 vs. 55.14, 55.07 vs. 55.47 and 54.45 vs. 54.62).

Table 9. Intent classification evaluation in terms of precision for the targeted data sets using an RNN and most important terms within KG using LIME (bold numbers indicate the best results for each setting; Orig. = original KG, Filt. = filtered KG; * denote statistically significant, p = 0.05).

ComQA Data Set	KG_t (100)		KG_t (500)		KG_t (750)		KG_{tr} (100)		KG_{tr} (500)		KG_{tr} (750)		KG_{tre} (100)		KG_{tre} (500)		KG_{tre} (750)	
Embeddings	Orig. 183	Filt. 86	Orig. 873	Filt. 315	Orig. 1,246	Filt. 392	Orig. 416	Filt. 115	Orig. 1272	Filt. 347	Orig. 1681	Filt. 425	Orig. 1510	Filt. 327	Orig. 2767	Filt. 590	Orig. 4133	Filt. 628
LASER+KG	89.74	89.10	91.54	90.26	90.49	90.49	90.03	**90.67** *	90.26	90.72	89.33	90.90	91.48	91.13	89.74	89.74	89.22	89.45
LASER+SBERT+KG	94.38	94.84	95.07	95.25	95.19	94.67	95.77	95.65	94.96	95.36	94.67	**95.25** *	94.78	95.30	94.78	95.48	94.78	94.61
LASER+MPNet+KG	94.84	95.19	95.19	94.90	94.72	94.26	95.13	94.61	94.03	94.61 *	93.91	93.91	94.49	94.38	92.93	93.39	93.16	94.20

ParaLex Data Set	KG_t (100)		KG_t (500)		KG_t (750)		KG_{tr} (100)		KG_{tr} (500)		KG_{tr} (750)		KG_{tre} (100)		KG_{tre} (500)		KG_{tre} (750)	
Embeddings	Orig. 169	Filt. 70	Orig. 785	Filt. 262	Orig. 1,116	Filt. 313	Orig. 343	Filt. 100	Orig. 1156	Filt. 313	Orig. 1473	Filt. 343	Orig. 641	Filt. 138	Orig. 1445	Filt. 340	Orig. 1,816	Filt. 388
LASER+KG	54.04	53.57	54.39	54.22	54.72	**55.14** *	53.94	54.15	54.74	55.14	54.48	**54.62** *	54.39	54.29	54.34	55.09	54.08	55.33
LASER+SBERT+KG	54.25	54.08	54.76	55.11	54.48	54.27	54.04	54.11	54.43	54.41	55.00	54.48	53.92	54.46	54.55	54.69	55.23	55.14
LASER+MPNet+KG	54.48	54.20	55.40	54.83	54.81	54.53	53.89	53.73	55.07	**55.47**	55.16	55.16	54.41	54.69	54.95	54.67	55.21	55.16

ProductServiceQA Data Set		KG_t		KG_{tr}		KG_{tre}	
Embeddings	Dimension	Orig. 136	Filt. 34	Orig. 494	Filt. 129	Orig. 1280	Filt. 286
LASER+KG	1324	63.64	63.51	63.16	62.42	63.42	63.16
LASER+SBERT+KG	2092	68.50	68.46	68.76	68.37	67.89	**68.86**
LASER+MPNet+KG	2092	69.60	68.94	69.03	**69.16**	68.77	68.16

Similarly to the aforementioned data sets, KG_{tr} and KG_{tre} improve the performance of the classification task over the original KGs generated on the **ProductServiceQA** data set (lower part of Table 9).

5.3.2. Filtering for Intent Classification with Siamese Networks

In addition to the RNN classification using the filtered KGs, we perform the same experiment with the Siamese network. As seen in the top part of Table 10 for the **ComQA** data set, filtered KG_t with 100 and 500 terms, respectively, significantly ($p < 0.05$) outperform the performance of the classification task in comparison to the original KGs, which contain a larger set of terms and relations.

Similarly, applying the filtered KGs to the **ParaLex** data set, KG_{tr} and KG_{tre} outperform their original counterparts while using the `SBERT` and `MPNet` embeddings, respectively.

As seen in the lower part of Table 10, the filtered KGs did not significantly outperform any of the original KGs generated from the **ProductServiceQA** data set. Nevertheless, minor improvement is detected for all KG variants with different SOTA embeddings.

Table 10. Intent classification evaluation in terms of precision for the targeted data sets using a Siamese network and most important terms within KG using LIME (bold numbers indicate the best results for each setting; Orig. = original KG, Filt. = filtered KG; * denote statistically significant, p = 0.05).

ComQA Data Set	KG_t (100)		KG_t (500)		KG_t (750)		KG_{tr} (100)		KG_{tr} (500)		KG_{tr} (750)		KG_{tre} (100)		KG_{tre} (500)		KG_{tre} (750)	
Embeddings	Orig. 183	Filt. 86	Orig. 873	Filt. 315	Orig. 1246	Filt. 392	Orig. 416	Filt. 115	Orig. 1272	Filt. 347	Orig. 1681	Filt. 425	Orig. 1510	Filt. 327	Orig. 2767	Filt. 590	Orig. 4133	Filt. 628
SBERT+KG	94.96	94.67	94.43	94.78	94.78	94.90	94.78	95.13	94.31	94.78	94.78	94.55	94.78	95.13	94.32	94.78	94.78	94.55
SBERT+LASER+KG	95.13	94.49	94.55	**95.25** *	94.60	94.84	94.78	94.96	94.55	94.55	94.95	94.66	94.14	94.61	93.80	94.32	92.93	92.12
MPNet+KG	95.13	**98.63** *	94.49	94.38	92.86	93.28	94.49	94.72	94.03	94.32	92.34	92.35	94.49	94.72	94.03	94.32	92.35	92.35
MPNet+LASER+KG	95.02	89.62	94.43	94.49	92.92	93.28	94.14	94.61	93.79	94.32	92.92	92.12	94.14	94.61	93.80	94.32	92.93	92.12
ParaLex Data Set	KG_t (100)		KG_t (500)		KG_t (750)		KG_{tr} (100)		KG_{tr} (500)		KG_{tr} (750)		KG_{tre} (100)		KG_{tre} (500)		KG_{tre} (750)	
Embeddings	Orig. 169	Filt. 70	Orig. 785	Filt. 262	Orig. 1116	Filt. 313	Orig. 343	Filt. 100	Orig. 1156	Filt. 313	Orig. 1473	Filt. 343	Orig. 641	Filt. 138	Orig. 1445	Filt. 340	Orig. 1816	Filt. 388
SBERT+KG	49.26	48.59	48.82	48.43	49.49	49.53	48.93	49.13	49.43	50.23	49.31	50.42	49.30	**50.14** *	50.59	50.54	49.91	49.91
SBERT+LASER+KG	49.17	48.94	49.52	48.59	49.17	49.86	48.65	48.83	49.28	50.40	49.35	50.28	52.72	52.04	53.66	53.10	52.77	52.44
MPNet+KG	52.29	50.35	51.21	52.11	51.54	52.28	50.18	52.16	50.55	52.79	51.86	**54.48** *	52.89	52.63	53.14	53.07	52.79	52.60
MPNet+LASER+KG	51.49	50.54	50.97	52.42	50.57	52.46	50.86	53.26	51.04	52.58	51.75	53.87	52.72	52.04	53.66	53.10	52.77	52.44

ProductServiceQA		KG_t		KG_{tr}		KG_{tre}	
Embeddings	Dimension	Orig. 136	Filt. 34	Orig. 494	Filt. 129	Orig. 1280	Filt. 286
SBERT+KG	1068	73.51	73.23	73.77	73.67	73.73	73.67
SBERT+LASER+KG	2092	73.16	73.06	74.08	74.06	73.07	**73.36**
MPNet+KG	1068	73.64	**73.80**	74.37	**74.50**	73.59	73.45
MPNet+LASER+KG	2092	73.81	73.49	73.77	73.14	73.29	73.49

5.4. Multilingual Setting

As a final experiment, we leverage the translations into English of the multilingual ProductServiceQA data set. Table 11 illustrates the intent classification task when the Spanish language is used. The best performance with pre-trained models is demonstrated with SOTA `MPNet+LASER` embeddings. When leveraging the KGEs trained on the automatically generated KGs, the classification precision increases to 65.57 when combining the embeddings as `MPNet+LASER+KG`.

Table 11. Intent classification evaluation for the Spanish ProductServiceQA data set translated into English using a Siamese network (bold numbers indicate the best results for each setting).

SOTA Embeddings	**Dimension**	**Precision**
SBERT	768	62.24
MPNET	768	64.34
SBERT+LASER	1092	61.62
MPNET+LASER	1092	**65.00**
Best Embeddings with KG	Dimension	Precision
MPNet+LASER+KG_t (100)	1392	**65.57**

We performed the same experiment with the Chinese intents, which were translated into English. In this setting, the `MPNet+LASER` embedding combination outperforms other SOTA pre-trained embeddings (Table 12). Similarly to the experiment on the Spanish language, employing the automatically generated KG, in this case in combination with `MPNet`, further improves the performance of the classification task.

Table 12. Intent classification evaluation for the Chinese ProductServiceQA data set translated into English using a Siamese network (bold numbers indicate the best results for each setting).

SOTA Embeddings	**Dimension**	**Precision**
SBERT	768	58.12
MPNET	768	59.30
SBERT+LASER	1092	59.01
MPNET+LASER	1092	**59.70**
Best Embeddings with KG	Dimension	Precision
MPNet+KG$_t$ (100)	1092	**60.66**

6. Conclusions

In this paper, we presented work on leveraging automatically generated knowledge graphs for intent classification. We provide an analysis of each step, i.e., term extraction, named entity recognition, and relation extraction, towards the creation of knowledge graphs 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 four different data sets and demonstrate that the domain-specific knowledge within knowledge graphs further improves the performance on the intent classification task. Furthermore, we study the set of terms and relations within the knowledge graphs and filter them by importance by leveraging the LIME tool. Finally, we leverage the Spanish and Chinese intents of the proprietary ProductServiceQA data set and leverage machine translation to perform the classification on noisy intents translated into English. Our ongoing work focuses on the use of knowledge graph extraction for use in multi-turn intent identification, more specifically on generating questions to direct a user to a more specific answer through knowledge subgraph identification.

Author Contributions: Conceptualization, S.D., H.A., J.P.M., and P.B.; methodology, M.A., S.D., H.A., J.P.M., and P.B.; software, S.M., C.R., G.V., D.P., and S.S.; validation, S.M., C.R., G.V., D.P., and S.S.; formal analysis, S.M., C.R., and G.V.; investigation, M.A. and G.V.; writing—original draft preparation, M.A., S.M., C.R., G.V., D.P., S.S., S.D., H.A., J.P.M., and P.B.; writing—review and editing, M.A., G.V., S.D., H.A., J.P.M., and P.B.; supervision, P.B.; project administration, M.A.; funding acquisition, P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant number SFI/12/RC/2289_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Extended intent classification evaluation for the ComQA data set using an RNN and the automatically extracted KGs (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Prec.	Best Embeddings with KG	Dim.	Prec.
SBERT	768	98.36	LASER+SBERT+KG$_t$ (500)	2092	**99.45**
LASER	1024	96.75	LASER+MPNet+KG$_t$ (750)	2092	**99.45**
MPNet	768	**98.63**	LASER+SBERT+KG$_t$ (750)/GloVe	2092	**99.45**
LASER+SBERT	1792	98.28			
LASER+SBERT+GloVe	2092	**98.63**			

			KG$_t$	KG$_t$	KG$_t$	KG$_{tr}$	KG$_{tr}$	KG$_{tr}$	KG$_{tre}$	KG$_{tre}$	KG$_{tre}$	
	Embeddings with KG	Dim.	(100)	(500)	(750)	(100)	(500)	(750)	(100)	(500)	(750)	DBpedia
	KG	300	40.71	75.41	86.89	45.08	75.13	83.61	79.96	84.70	93.34	14.92
Concat.	LASER+KG	1324	95.35	95.62	95.08	95.63	95.08	95.08	95.90	95.90	95.63	96.17
	LASER+SBERT+KG	2092	98.90	99.18	**99.45**	98.91	98.63	98.63	98.36	98.63	98.91	98.91
	LASER+MPNet+KG	2092	99.18	**99.45**	98.09	98.91	98.36	98.63	98.09	98.63	98.36	98.36
Substit.	LASER+KG/GloVe	1324	94.81	94.54	95.36	94.81	93.72	94.26	95.36	96.72	95.36	96.72
	LASER+SBERT+KG/GloVe	2092	98.36	98.63	98.91	98.09	98.91	**99.45**	98.91	98.91	98.36	98.09
	LASER+MPNet+KG/GloVe	2092	97.54	98.09	98.36	97.54	98.36	98.09	98.36	98.91	97.81	98.36

Table A2. Extended intent classification evaluation for the ParaLex data set using an RNN and the automatically extracted KGs (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision	Best Embeddings with KG	Dim.	Precision
SBERT	768	54.06	LASER+MPNet+KG/GloVe	2092	**55.42**
LASER	1024	52.92			
MPNet	768	53.80			
LASER+SBERT	1792	54.07			
LASER+SBERT+GloVe	2092	**54.41**			

			KG$_t$	KG$_t$	KG$_t$	KG$_{tr}$	KG$_{tr}$	KG$_{tr}$	KG$_{tre}$	KG$_{tre}$	KG$_{tre}$	
	Embeddings with KG	Dim.	(100)	(500)	(750)	(100)	(500)	(750)	(100)	(500)	(750)	DBpedia
	KG		22.38	46.67	49.39	25.86	47.82	47.65	30.34	48.69	50.45	20.15
Concat.	LASER+KG	1324	54.04	54.39	54.72	53.94	54.74	54.48	54.43	54.95	54.46	53.24
	LASER+SBERT+KG	2092	54.25	54.76	54.48	54.04	54.43	55.00	54.11	54.67	54.29	53.66
	LASER+MPNet+KG	2092	54.48	55.40	54.81	53.89	55.07	55.16	54.46	55.28	55.14	53.66
Substit.	LASER+KG/GloVe	1324	51.41	54.27	53.47	52.91	54.20	54.27	54.25	54.46	54.29	51.55
	LASER+SBERT+KG/GloVe	2092	52.37	54.39	53.26	52.11	52.49	53.54	54.58	54.90	55.16	53.43
	LASER+MPNet+KG/GloVe	2092	51.69	54.65	53.10	53.45	53.40	54.79	54.62	55.35	**55.42**	51.64

Table A3. Extended intent classification evaluation for the ProductServiceQA data set using an RNN and the automatically extracted KG with 100 terms (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision	Best Embeddings with KG	Dim.	Precision
SBERT	768	68.02	LASER+MPNet+DBpedia	2092	**70.00**
LASER	1024	62.68			
MPNet	768	**69.25**			
LASER+SBERT	1792	68.60			
LASER+SBERT+GloVe	2092	68.40			

	Embeddings with KG	Dim.	Bench. KG$_t$	Bench. KG$_{tr}$	Bench. KG$_{tre}$	KG$_t$	KG$_{tr}$	KG$_{tre}$	KG$_t ref$	DBpedia
	KG	300	26.19	34.91	38.10	25.62	31.80	45.15	39.33	23.61
Concat.	LASER+KG	1324	63.20	62.06	62.46	63.64	63.16	63.42	63.03	62.77
	LASER+SBERT+KG	2092	68.68	68.37	67.14	68.50	68.76	67.89	68.11	67.37
	LASER+MPNet+KG	2092	68.77	68.94	68.24	69.51	68.16	68.77	69.21	**70.00**
Substit.	LASER+KG/GloVe	1324	59.75	61.76	60.93	59.75	60.18	62.33	62.07	60.27
	LASER+SBERT+KG/GloVe	2092	67.15	67.85	68.33	67.76	68.55	68.46	68.07	67.76
	LASER+MPNet+KG/GloVe	2092	67.59	67.02	66.14	67.85	68.51	67.15	68.37	68.64

Table A4. Extended intent classification evaluation for the ATIS data set using an RNN and the automatically extracted KG with 100 terms (bold numbers indicate the best results for each setting).

	SOTA Embeddings	Dim.	Precision	Best Embeddings with KG	Dim.	Precision
	SBERT	768	98.67	LASER+KG	1324	**99.25**
	LASER	1024	**98.87**	LASER+MPNet+KG	2092	**99.25**
	MPNet	768	98.43	LASER+SBERT+KG/GloVe	2092	**99.25**
	LASER+SBERT	1792	98.50	LASER+MPNet+KG/GloVe	2092	**99.25**
	LASER+SBERT+GloVe	2092	98.62			
	Embeddings with KG	Dim.	KG_t	KG_{tr}		KG_{tre}
	KG	300	91.37	91.37		93.87
Concat.	KG	300	91.37	91.37		93.87
	KG+Glove	600	98.25	98.62		98.00
	LASER+KG	1324	**99.25**	98.62		98.25
	LASER+SBERT+KG	2092	98.25	**99.25**		98.50
	LASER+MPNet+KG	2092	**99.25**	**99.25**		98.50
Substit.	LASER+KG/GloVe	1324	98.87	98.62		97.87
	LASER+SBERT+KG/GloVe	2092	99.00	98.37		**99.25**
	LASER+MPNET+KG/GloVe	2092	99.12	99.12		**99.25**

Table A5. Extended intent classification evaluation for the ComQA data set using a Siamese network and the automatically extracted KGs (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision			Best Embeddings with KG		Dim.	Precision		
SBERT	768	**95.18**			SBERT+LASER+KG		2092	**95.18**		
SBERT+LASER	1792	94.66								
MPNET	768	94.37								
MPNET+LASER	1792	94.14								
		KG_t	KG_t	KG_t	KG_{tr}	KG_{tr}	KG_{tr}	KG_{tre}	KG_{tre}	KG_{tre}
Embeddings with KG	Dim.	(100)	(500)	(750)	(100)	(500)	(750)	(100)	(500)	(750)
SBERT+KG	1068	94.96	94.43	94.78	94.78	94.31	94.78	94.55	94.61	94.78
SBERT+LASER+KG	2092	95.13	94.55	94.60	94.78	94.55	94.95	**95.18**	94.55	94.43
MPNet+KG	1068	95.13	94.49	92.86	94.49	94.03	92.34	93.43	92.87	92.23
MPNet+LASER+KG	2092	95.02	94.43	92.92	94.14	93.79	92.92	93.91	93.27	91.65

Table A6. Extended intent classification evaluation for the ComQA data set, filtered by questions with a frequency between two and five, using a Siamese network and the automatically extracted KGs (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision			Best Embeddings with KG		Dim.	Precision		
SBERT	768	83.31			SBERT+LASER+KG		2092	**84.87**		
SBERT+LASER	1792	84.12								
MPNET	768	83.25								
MPNET+LASER	1792	**84.23**								
		KG_t	KG_t	KG_t	KG_{tr}	KG_{tr}	KG_{tr}	KG_{tre}	KG_{tre}	KG_{tre}
Embeddings with KG	Dim.	(100)	(500)	(750)	(100)	(500)	(750)	(100)	(500)	(750)
SBERT+KG	1068	83.78	83.48	84.24	84.07	84.65	83.95	84.01	83.31	83.60
SBERT+LASER+KG	2092	**84.87**	84.42	83.54	83.31	83.89	84.01	84.18	83.14	83.78
MPNet+KG	1068	84.29	84.07	84.12	84.30	83.37	83.89	83.95	83.89	83.95
MPNet+LASER+KG	2092	83.02	83.89	83.49	83.89	83.31	83.89	83.54	84.36	83.78

Table A7. Extended intent classification evaluation for the ParaLex data set using a Siamese network and the automatically extracted KGs (bold numbers indicate the best results for each setting; * denote statistically significant, p = 0.05).

SOTA Embeddings	Dim.	Precision	Best Embeddings with KG	Dim.	Precision
SBERT	768	48.81	MPNet+KG	1068	**52.29**
SBERT+LASER	1792	49.75			
MPNET	768	50.33			
MPNET+LASER	1792	**50.47**			

		KG_t	KG_t	KG_t	KG_{tr}	KG_{tr}	KG_{tr}	KG_{tre}	KG_{tre}	KG_{tre}
Embeddings with KG	Dim.	(100)	(500)	(750)	(100)	(500)	(750)	(100)	(500)	(750)
SBERT+KG	1068	49.26	48.82	49.49	48.93	49.43	49.31	49.28	49.73	49.12
SBERT+LASER+KG	2092	49.17	49.52	49.17	48.65	49.28	49.35	48.89	48.93	49.49
MPNet+KG	1068	**52.29** *	51.21	51.54	50.18	50.55	51.86	51.18	51.68	50.62
MPNet+LASER+KG	2092	51.49	50.97	50.57	50.86	51.04	51.75	51.28	50.69	51.20

Table A8. Extended intent classification evaluation for the ProductServiceQA data set using a Siamese network and the automatically extracted KGs with 100 terms (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision	Best Embeddings with KG	Dim.	Precision
SBERT	768	**73.94**	MPNet+LASER+KG	2092	**74.69**
SBERT+LASER	1792	73.77			
MPNET	768	73.55			
MPNET+LASER	1792	73.51			

Embeddings with KG	Dim.	Bench. KG_t	Bench. KG_{tr}	Bench. KG_{tre}	KG_t	KG_{tr}	KG_{tre}
SBERT+KG	1068	74.03	73.73	74.56	73.51	73.77	73.73
SBERT+LASER+KG	2092	73.68	73.77	73.51	73.16	74.08	73.07
MPNet+KG	1068	74.08	73.64	73.68	73.64	74.37	73.59
MPNet+LASER+KG	2092	74.64	**74.69**	74.16	73.81	73.77	73.29

Table A9. Extended intent classification evaluation for the ATIS data set using a Siamese network and the automatically extracted KGs with 100 terms (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision	Best Embeddings with KG	Dim.	Precision
SBERT	768	**99.37**	MPNet+KG	1068	**99.50**
SBERT+LASER	1792	99.00			
MPNET	768	98.62			
MPNET+LASER	1792	98.62			

Embeddings with KG	Dim.	KG_t	KG_{tr}	KG_{tre}
SBERT+KG	1068	99.25	99.50	99.37
SBERT+LASER+KG	2092	99.00	99.37	99.25
MPNet+KG	1068	99.12	99.12	**99.50**
MPNet+LASER+KG	2092	98.75	98.50	99.37

Table A10. Extended intent classification evaluation for the Spanish ProductServiceQA data set translated into English using a Siamese network and the automatically extracted KGs with 100 terms (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision		Best Embeddings with KG		Dim.	Precision
SBERT	768	62.24		MPNet+LASER+KG			**65.57**
MPNET	768	64.34					
SBERT+LASER	1792	61.62					
MPNET+LASER	1792	**65.00**					
Embeddings with KG	Dim.	Bench. KG_t	Bench. KG_{tr}	Bench. KG_{tre}	KG_t	KG_{tr}	KG_{tre}
SBERT+KG	1068	62.89	61.58	61.76	62.33	62.41	62.54
SBERT+LASER+KG	2092	62.33	61.97	62.94	61.32	62.46	60.92
MPNet+KG	1068	63.95	60.00	61.06	64.34	63.90	60.31
MPNet+LASER+KG	2092	**65.57**	59.13	62.41	64.91	63.29	59.83

Table A11. Extended intent classification evaluation for the Chinese ProductServiceQA data set translated into English using a Siamese network and the automatically extracted KGs with 100 terms (bold numbers indicate the best results for each setting).

SOTA Embeddings	Dim.	Precision		Best Embeddings with KG		Dim.	Precision
SBERT	768	58.12		MPNet+KG			**60.66**
MPNET	768	59.30					
SBERT+LASER	1792	59.01					
MPNET+LASER	1792	**59.70**					
Embeddings with KG	Dim.	Bench. KG_t	Bench. KG_{tr}	Bench. KG_{tre}	KG_t	KG_{tr}	KG_{tre}
SBERT+KG	1068	58.47	58.25	58.56	58.08	59.70	58.12
SBERT+LASER+KG	2092	57.32	57.42	58.51	58.30	57.42	58.34
MPNet+KG	1068	60.57	55.71	56.15	**60.66**	59.48	55.54
MPNet+LASER+KG	2092	60.18	55.23	57.59	60.40	58.82	55.45

References

1. Temerak, M.S.; El-Manstrly, D. The influence of goal attainment and switching costs on customers' staying intentions. *J. Retail. Consum. Serv.* **2019**, *51*, 51–61. https://doi.org/10.1016/j.jretconser.2019.05.020.
2. Abujabal, A.; Roy, R.S.; Yahya, M.; Weikum, G. ComQA: A Community-sourced Dataset for Complex Factoid Question Answering with Paraphrase Clusters. *arXiv* **2018** arXiv:1809.09528.
3. Fader, A.; Zettlemoyer, L.; Etzioni, O. Paraphrase-Driven Learning for Open Question Answering. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013.
4. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; et al. DBpedia—A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semant. Web J.* **2015**, *6*, 167–195.
5. Cavalin, P.; Alves Ribeiro, V.H.; Appel, A.; Pinhanez, C. 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), Online, 16–20 November 2020; pp. 3952–3961. https://doi.org/10.18653/v1/2020.emnlp-main.324.
6. Zhang, H.; Zhang, Y.; Zhan, L.M.; Chen, J.; Shi, G.; Wu, X.M.; Lam, A.Y. Effectiveness of Pre-training for Few-shot Intent Classification. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 16–20 November 2021; pp. 1114–1120.
7. Zhang, J.; Ye, Y.; Zhang, Y.; Qiu, L.; Fu, B.; Li, Y.; Yang, Z.; Sun, J. Multi-Point Semantic Representation for Intent Classification. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 9531–9538. https://doi.org/10.1609/aaai.v34i05.6498.
8. Purohit, H.; Dong, G.; Shalin, V.; Thirunarayan, K.; Sheth, A. Intent Classification of Short-Text on Social Media. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 222–228. https://doi.org/10.1109/SmartCity.2015.75.
9. Ahmad, Z.; Ekbal, A.; Sengupta, S.; Maitra, A.; Ramnani, R.; Bhattacharyya, P. Unsupervised Approach for Knowledge-Graph Creation from Conversation: The Use of Intent Supervision for Slot Filling. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534398.
10. Yu, C.; Wang, W.; Liu, X.; Bai, J.; Song, Y.; Li, Z.; Gao, Y.; Cao, T.; Yin, B. FolkScope: Intention Knowledge Graph Construction for Discovering E-commerce Commonsense. *arXiv* **2022**, arXiv:2211.08316.

11. Pinhanez, C.S.; Candello, H.; Cavalin, P.; Pichiliani, M.C.; Appel, A.P.; Alves Ribeiro, V.H.; Nogima, J.; de Bayser, M.; Guerra, M.; Ferreira, H.; et al. Integrating Machine Learning Data with Symbolic Knowledge from Collaboration Practices of Curators to Improve Conversational Systems. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 8–13 May 2021; Association for Computing Machinery: New York, NY, USA, 2021.
12. He, Y.; Jia, Q.; Yuan, L.; Li, R.; Ou, Y.; Zhang, N. A Concept Knowledge Graph for User Next Intent Prediction at Alipay. *arXiv* **2023**, arXiv:2301.00503. https://doi.org/10.48550/arXiv.2301.00503.
13. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1441–1451. https://doi.org/10.18653/v1/P19-1139.
14. He, T.; Xu, X.; Wu, Y.; Wang, H.; Chen, J. Multitask Learning with Knowledge Base for Joint Intent Detection and Slot Filling. *Appl. Sci.* **2021**, *11*, 4887. https://doi.org/10.3390/app11114887.
15. Siddique, A.B.; Jamour, F.T.; Xu, L.; Hristidis, V. Generalized Zero-shot Intent Detection via Commonsense Knowledge. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, 11–15 July 2021.
16. Shabbir, J.; Arshad, M.U.; Shahzad, W. NUBOT: Embedded Knowledge Graph With RASA Framework for Generating Semantic Intents Responses in Roman Urdu. *arXiv* **2021**, arXiv:2102.10410.
17. Sant'Anna, D.T.; Caus, R.O.; dos Santos Ramos, L.; Hochgreb, V.; dos Reis, J.C. Generating Knowledge Graphs from Unstructured Texts: Experiences in the E-commerce Field for Question Answering. In Proceedings of the Joint Proceedings of Workshops AI4LEGAL2020, NLIWOD, PROFILES 2020, QuWeDa 2020 and SEMIFORM2020 Colocated with the 19th International Semantic Web Conference (ISWC 2020), Virtual Conference, Athens, Greece, 1–6 November 2020 ; Koubarakis, M., Alani, H., Antoniou, G., Bontcheva, K., Breslin, J.G., Collarana, D., Demidova, E., Dietze, S., Gottschalk, S., Governatori, G., et al., Eds.; CEUR-WS.org, 2020; Volume 2722, *CEUR Workshop Proceedings*, pp. 56–71.
18. Hu, J.; Wang, G.; Lochovsky, F.; Sun, J.t.; Chen, Z. Understanding User's Query Intent with Wikipedia. In Proceedings of the 18th International Conference on World Wide Web, WWW '09, Madrid, Spain, 20–24 April 2009; p. 471–480. https://doi.org/10.1145/1526709.1526773.
19. Balažević, I.; Allen, C.; Hospedales, T.M. TuckER: Tensor Factorization for Knowledge Graph Completion. In Proceedings of the Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019.
20. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162.
21. Artetxe, M.; Schwenk, H. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 597–610. https://doi.org/10.1162/tacl_a_00288.
22. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics.
23. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. MPNet: Masked and Permuted Pre-training for Language Understanding. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020 ; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc., Red Hook, NY, USA: 2020; Volume 33, pp. 16857–16867.
24. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
25. Mcnemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157.
26. Bordea, G.; Buitelaar, P.; Polajnar, T. Domain-independent term extraction through domain modelling. In Proceedings of the 10th International Conference on Terminology and Artificial Intelligence (TIA 2013), Paris, France, 2013.
27. Manjunath, S.H.; McCrae, J.P. Encoder-Attention-Based Automatic Term Recognition (EA-ATR). In Proceedings of the 3rd Conference on Language, Data and Knowledge (LDK 2021), Zaragoza, Spain, 1–3 September 2021; Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Dagstuhl, Germany: 2021.
28. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual String Embeddings for Sequence Labeling. In Proceedings of the COLING 2018, 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1638–1649.
29. Pereira, B.; Robin, C.; Daudert, T.; McCrae, J.P.; Mohanty, P.; Buitelaar, P. Taxonomy Extraction for Customer Service Knowledge Base Construction. In *Semantic Systems. The Power of AI and Knowledge Graphs*; Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 175–190.
30. Chen, D.; Manning, C. A Fast and Accurate Dependency Parser using Neural Networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 740–750. https://doi.org/10.3115/v1/D14-1082.
31. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010.

32. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings; Bengio, Y.; LeCun, Y., Eds.; 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.