Towards One-Shot Learning for Text Classification using Inductive Logic Programming

Ghazal Afroozi Milani, Daniel Cyrus, Alireza Tamaddoni-Nezhad

Department of Computer Science, University of Surrey, United Kingdom

{g.afroozimilani,d.cyrus,a.tamaddoni-nezhad}@surrey.ac.uk

With the ever-increasing potential of AI to perform personalised tasks, it is becoming essential to develop new machine learning techniques which are data-efficient and do not require hundreds or thousands of training data. In this paper, we explore an Inductive Logic Programming approach for one-shot text classification. In particular, we explore the framework of Meta-Interpretive Learning (MIL), along with using common-sense background knowledge extracted from ConceptNet. Results indicate that MIL can learn text classification rules from a small number of training examples, even one example. Moreover, the higher complexity of chosen example for one-shot learning, the higher accuracy of the outcome. Finally, we utilise two approaches, Background Knowledge Splitting and Average One-Shot Learning, to evaluate our model on a public News Category dataset. The outcomes validate MIL's superior performance to the Siamese net for one-shot learning from text.

1 Introduction

Machine learning, in particular, Deep Neural Networks (DNNs), has achieved outstanding outcomes in numerous real world applications such as image and text classification [7, 20]. However, these learning methods require a huge number of training instances, which is not always possible to be provided in advance. An automated software tool, for example, should be able to learn from a small number of interactions with the user in order to be efficiently customised with every new user's requirement. Inductive Logic Programming (ILP) and, in particular, Meta Interpretive Learning (MIL) can learn human-readable hypotheses from a small amount of training data [17, 24]. This capability is very promising for medical and industrial usage, especially when we do not have access to a large amount of training data. The main contribution of this study is to develop and evaluate a new learning algorithm, Meta-Goal Learner (MGL), based on the framework of Meta-Interpretive Learning (MIL), which can be utilised for one-shot learning from textual data along with using a common-sense knowledge-base (i.e. ConceptNet) as background knowledge for learning.

2 Inductive Logic Programming

Inductive Logic Programming (ILP) is a subfield of symbolic artificial intelligence, which works based on inductive reasoning to detect generalisations, rules, or models [13, 18]. In this method, the system learns from training examples with the help of background information and using logic programming [24]. Indeed, the primary purpose of ILP [13], like the other types of machine learning, is to induce a model/ hypothesis that can generalise training examples but, unlike them, uses logic programs to represent data and learns relations [2].

Using logic programs brings several benefits to ILP. Firstly, as they represent hypotheses in ILP and are related to relational databases, they can support relational data like graphs. On the other hand, because of its expressivity, it can learn complex relational theories like cellular automata, event calculus theories, and Petri nets [2]. Secondly, it is similar to natural language; therefore, readable and valuable for explainable AI and scientific discovery ([17]).

Every standard ILP setting consists of three input sets, including Background Knowledge (B), positive training examples (E^+), and negative training examples (E^-) [11, 13]. The output is a Hypothesis H such that:

- It is necessary $\forall\, e \in E^+$ $B \not\models e$

- H is complete/ sufficient if $\forall\, e \in E^+$ and $H \wedge B \models e$

- H is strong consistent if $\forall\, e \in E^-$ and $H \wedge B \not\models e$

- H is weak consistent if $H \wedge B \not\models \square$

2.1   Meta-Interpretive Learning

Meta-Interpretive Learning (MIL) [15] is a type of Inductive Logic Programming which allows learning logic programs from background knowledge, training examples and a declarative bias called metarules. Metarules are datalog clauses with variables quantified over predicate symbols (i.e., second-order variables) [19, 10].

The advantage of MIL is its capability to (1) automatically introduce new sub-definitions when it learns a predicate definition [16, 15]. This is referred to as Predicate Invention. (2) learning recursive clauses. MIL uses metarules to restrict hypotheses space. In other words, a meta-rule is a higher-order expression in which the form of permitted clauses in the hypothesis space is instructed[16, 12].

A **metarule** is a higher-order well formed formula as follows:
$$\exists\sigma\forall\tau P(s_1,...,s_m) \leftarrow ...,Q_i(t_1,...,t_n),...$$
where σ, τ are disjoint variables, ρ and ζ are sets of predicate symbols and constants respectively $P,Q_i \in \sigma \cup \tau \cup \rho$ and $s_1,...,s_m,t_1,...,t_n \in \sigma \cup \tau \cup \zeta$. Metarules are shown normally without quantifiers as: $P(s_1,...,s_m) \leftarrow ...,Q_i(t_1,...,t_n),...$. P and Q_i are existentially quantified higher-order variables, while the other variables present universally quantified variables.

The standard MIL problem [15] is defined as follows: Given metarules (M), definite program background knowledge (B) and ground positive and negative unit examples (E^+, E^-), **MIL** returns a higher-order datalog program hypothesis (H) if one exists such that: $M \wedge B \wedge H \models E^+$ and $M \wedge B \wedge H \not\models E^-$ is consistent.

In ILP, selecting proper Background Knowledge (B) is essential for appropriate learning performance. However, this could be difficult or expensive as often this should be done by experts. There are two main problems related to having a balance background knowledge, including too little and too much Background Knowledge. If using too little Background Knowledge, we might lose the target hypothesis. Having too much irrelevant Background Knowledge is another ILP challenge that can increase hypothesis space size and decrease learning performance ([4]).

Category	Source	Relations	Example
Commonsense KGs	ConceptNet	37	mother-related_to-family
	Atomic	9	Person X bakes bread-xEffect-eat food
	Glucose	10	someone A -eats- something A
	WebChild	4groups	eating -type of- consumption
	Quasimodo	78636	pressure cooker-cook faster-food
	SenticNet	1	cold food -polarity- negative
	HasPartKB	1	dairy food -has part- vitamin
	Probase	1	apple -is a- food
	Isacore	1	snack food -is a- food
Common KGs	Wikidata	6.7k	food -has quality- mouthfeel
	YAGO4	116	banana chip -rdf:type- food
	DOLCE	1	n/p
	SUMO	1614	food -hyponym- food_product
Lexical resources	WordNet	10	food -hyponym- comfort food
	Roget	2	dish -synonym- food
	FrameNet	8(f2f)	eating -evoke- Ingestion
	MetaNet	14(f2f)	food -has role- food_consumer
	VerbNet	36(roles)	eating -haspatient- comestible
Visual sources	Visual Genome	42374	food -on- plate
	Flickr30k	1	eating place -confers with- their kitchen
Corpora& LMs	GenericsKB	n/a	Animals search for food
	GPT-2	n/a	eating at home will not lead to weight gain
	ChatGPT	n/a	Food provides nourishment: People eat food to.. (huge amount of information about eating)

Table 1: Overview of common-sense knowledge sources partly taken from [6]

There are two methods to overcome this issue, one is to enable an ILP system to invent a new predicate instead of missing background knowledge [23]. The second method is to use transfer learning to discover knowledge that can help mitigate the effects of background knowledge shortage [8].

3   Commonsense Knowledge and ConceptNet

Commonsense knowledge is an ordinary information that helps people make sense of everyday situations without mentioning them in their communications. Due to its implicitness, capturing this knowledge is beneficial for designing an effective human-computer interface and various types of Artificial Intelligence (AI) [6, 5]. These common-sense knowledge sources are in various forms and about different types of knowledge. Ilievski et al. [6] divided them into five categories, common-sense knowledge graphs (e.g. ConceptNet and Atomic), common knowledge graphs and ontologies(e.g. Wikidata and Yago), lexical resources (e.g. WordNet and Roget), visual common-sense sources (e.g. Visual Genome), and Corpora and language models (e.g. GenericsKB and Language models) as shown in Table 1.

ConceptNet is a multilingual common-sense knowledge graph derived from the crowd-sourced Open Mind Common Sense project and knowledge from other resources like WordNet, DBPedia, Games with
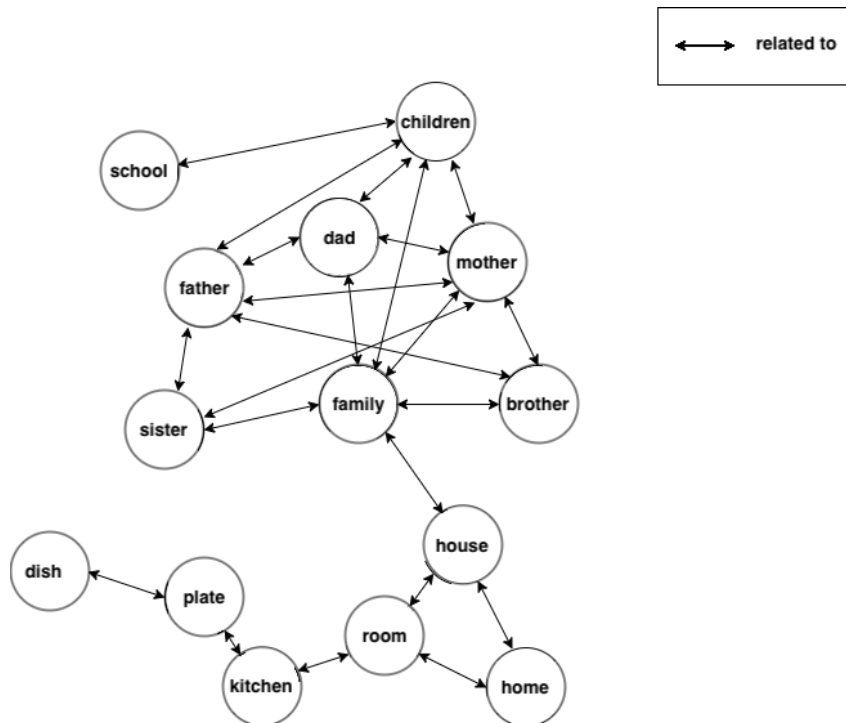
Figure 1: An illustration of ConceptNet used by our system.

a purpose, Wiktionary, OpenCyc and JMDict, connecting words and terms of natural language as nodes with labelled weighted edges with 37 types of relations [6].

The standard definition of a Common-sense Knowledge Graph is as follows [26]:
G= (V,E,R) is a Common-sense Knowledge Graph, where V, E and R are the node, edge and relation sets. Edges comprise triplets (h, r, t) where h and t are head and tail entities connected by relation $r : E = (h, r, t) \mid h \in V, t \in V, r \in R$, and nodes are along with a free-text description.

These node relations in ConceptNet5.5 have been categorised [21] to two sub-classes: Symmetric relations like LocatedNear, RelatedTo, SimilarTo, and Asymmetric relations such as AtLocation, Capa-bleOf, CreatedBy, DefinedAs, HasA, and Entails [6]. We use relations of "RelatedTo", which demon-strate an undefined meaning relation between two words. As shown in Figure 1, every node represents a word extracted from ConceptNet and connected to another node by a bidirectional edge. These symmet-ric edges show the relatedness of the connected words and, therefore, are bidirectional.

4   One-shot text classification using ILP and MIL

The basis of an interactive learning system is to learn from the user. This challenge requires incremental learning techniques that help the system to learn from time to time. In this section, we introduce Meta-Goal Learner (MGL), a novel machine learning from user interactions. We explore Metagol[3] learning engine for the task of few-shot learning in MGL.

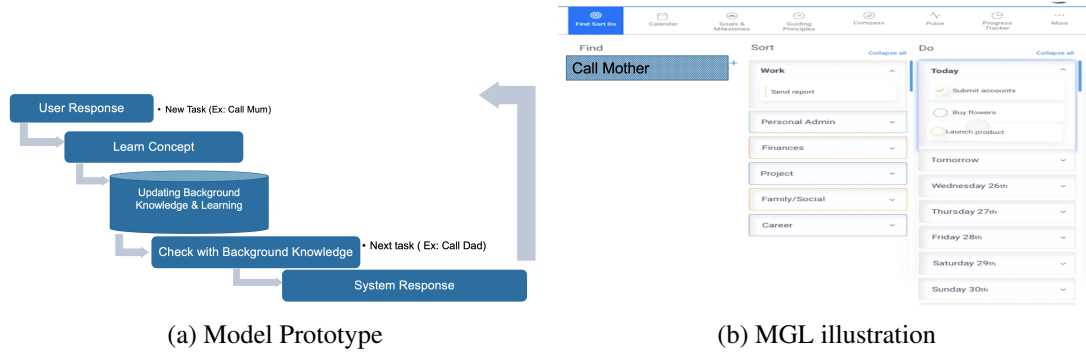(a) Model Prototype   (b) MGL illustration

Figure 2: Meta-Goal Learner.

MGL initially fetches the words of the task (i.e. sentence given by the user) and obtains relevant background knowledge from ConceptNet. The system can then learn and categorise the upcoming tasks, as shown in Figure 2. For instance, when a user writes a task (e.g. "call mother"), the system will wait for the first answer/ label from the user and then will consider it as a positive example. In this case, the positive example represents the task and its category name. The system then generates relevant background knowledge as follows:

1. Tokenise the sentence and generate clauses as part of the background knowledge:
   e.g., contains(X,call). contains(X,mum).

2. Send a request to ConceptNet web API for fetching the relations of the words separately and adding them to the background knowledge:
   e.g., related_to(call, phone). related_to(mother, family).

Given the positive example and background knowledge, the system will learn the hypothesis using ILP. When the user adds the next sentence, the system will follow steps one and two and checks with the hypotheses learned before. If there is a match, the system will label the sentence as before; otherwise, the system will try to learn a new classification rule using the current input task as a positive example and examples from other tasks as negative examples. In the next section, we present some of the experimental evaluations of the learning algorithms which we have been exploring for integration within MGL.

4.1 ILP Experiments using the learning engine Aleph

In this experiment, we check whether it is possible to use the ILP learning engine Aleph[22] for few-shot text classification. Examples and background knowledge are defined as follows:

Examples We use two positive examples for each of the "family", "work", and "sport" categories. For negative examples, we use every example of each category as the negative example for the other category.

Background Knowledge To provide Background knowledge, we initially omit stop-words using NLTK library. Then, the system generates predicates for each sentence in two steps:

1. Indicating every word of the sentences in a "contains" predicate
   (e.g. contains([registering,gym],gym)).

2. Sending a request to ConceptNet API for each word of the sentence to fetch their "related" terms and then generating their "related_to" predicate (e.g. related_to(mother,family)).

Results and discussion Using the learning engine Aleph, three rules (hypotheses) have been generated to classify the three categories.

Rule 1: category(A,family) :- contains(A,B), related_to(B,shop).
Rule 2: category(A,work) :- contains(A,B), related_to(B,letter).
Rule 3: category(A,sport) :- contains(A,B), related_to(B,exercise).

As explained above the training data for each category included 2 positive examples. We also experimented with one positive example (one-shot learning) but Aleph was unable to learn any rule. To evaluate the accuracy of the above rules on separate test data, we use a different dataset with 29 positive examples, including 15 "family", 9 "work" and 5 "sport" examples and the negative test data for each group consists of examples from other groups (i.e., 15 family examples used as negative for work, 9 work examples are used as negative for sport, and 5 sport examples are used as negative for family) and reached to the overall predictive accuracy of 64%.

4.2   MIL Experiments using the learning engine Metagol

In this experiment, we first check whether it is possible to learn a hypothesis from one positive and one negative example using MIL then explore the possibility of learning more general rules using predicate invention and recursive rules in MIL.

We employ the learning engine Metagol [14] to utilise MIL with the same Background Knowledge and examples applied in the Aleph experiment. In addition to training examples and background knowledge, Metagol also needs metarules to induce the hypothesis. We test our system with two different types of metarules. At first, we utilise a *chain* metarule to obtain the learning hypothesis. In the second experiment, we use *indent*, *chain*, and *recursive* metarules along with using constants.

Results and discussion The system induces one hypothesis from one positive and one negative example. Note that in this experiment, the positive example is a simple case where the category name is directly related to one of the words in the sentence. Consequently, one chain metarule is enough to generate the hypothesis. For instance, in category([call, mother], family), 'mother' directly connects to 'family' with only one edge of 'related_to' relation in ConceptNet. Hence, the acquired hypothesis consists of one 'related_to' predicate as follows:

category(A,B) :- contains(A,C), related_to(C,B).

This rule is similar to the rule which was learned by Aleph in the previous section when we provided at least two positive examples for each category and used variable in mode declaration but Aleph (unlike Metagol) was unable to learn from just one positive example. However, the rule above cannot cover all relevant positive examples as it only considers 'related_to' once. To demonstrate this, we provided more training examples (4 positive and 6 negative) which include cases where we need at least two levels of connections, along with using chain metarule. Consequently, the system invents a new predicate (i.e.,

category_1(A,B):- related_to(A,C),related_to(C,B).) within its hypothesis set as follows:

category(A,B) :- contains(A,C), category_1(C,B).
category_1(A,B):- related_to(A,C),related_to(C,B).

The new rule can capture cases such as the category name 'home' which does not directly connect to 'mother' but is connected via 'family'. However, in order to have a general transitive rule for 'related_to' we need to learn a recursive rule.

Finally, we demonstrate that Metagol can be used to learn a recursive rule to capture transitive 'related_to' relations. In this experiment, we provide two positive examples with several relatednesses in their structure. Moreover, we use constant and variable at the indent metarule and consequently, the following results including a complete recursive hypothesis are achieved:

category(A,B):-contains(A,C),category_1(C,B).
category_1(A,B):-related_to(A,C),category_1(C,B).
category_1(A,home):-related_to(A,home).

As shown above, it is notable that whatever a chosen example for one-shot learning has more relatedness in its structure, i.e. higher complexity compared to the other examples, it could achieve a complete recursive hypothesis. This, in turn, is especially beneficial for choosing the best possible shot to do one-shot learning.

To evaluate one-shot learning, we employ the same test dataset used for the experiments with Aleph in the previous section. As evidenced by the results, Metagol could achieve around 70% accuracy, while Aleph could not learn any rule from one example.

In summary, we demonstrated that the Metagol learning engine can be used in our system to learn text classification rules from one example (one-shot learning). This feature clearly distinguishes Metagol (MIL) from Aleph (standard ILP). Moreover, we also demonstrated learning recursive rules in addition to predicate invention. However, this capability required metarules defined for the given application. In future, we will explore new implementations of MIL which can automatically generate the required metarules from background knowledge and examples.

4.3 Experiments comparing MIL with Deep Learning

In this experiment, we use Siamese Network for few-shot learning for text classification and compare the results with MIL from the previous sections. A Siamese network includes two identical sub-networks with shared weights. Each sub-network works simultaneously and compares its outputs at the end [1]. If the input of the twin sub-networks is the same, they extract similar semantic features, and the distance between their output will be less; otherwise, the distance will be more significant. The Siamese Network used in this experiment is similar to the CNN-based Siamese model described in [27] where each twin sub-networks consist of a convolutional layer, a max pooling layer and a fully connected layer. The model sets with a batch size of 50, and a learning rate of 0.00001. To achieve optimal results, the network is trained over ten epochs.

Following the approach described in [25], we evaluate the performance of the Siamese network and compare the results with our one-shot learning approach using Metagol. The training data for both systems are one positive example and one or more negative examples (examples from other categories) and the test data is the same as the one used in the previous section.

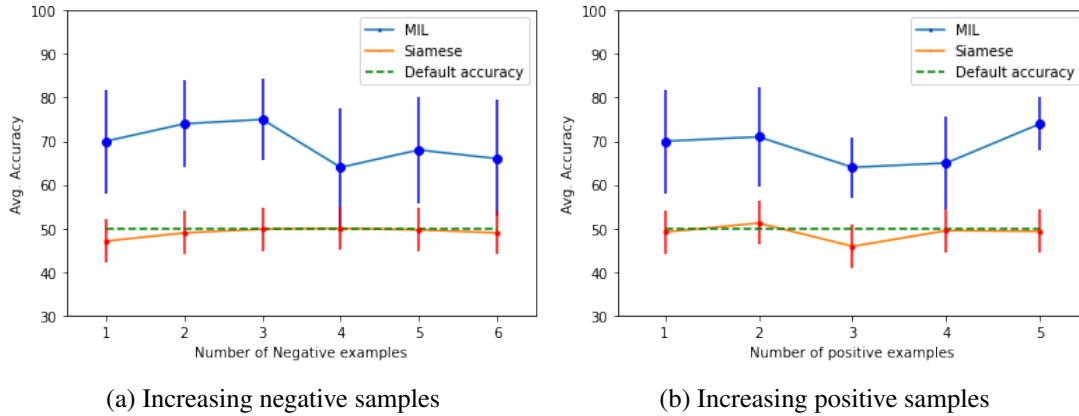(a) Increasing negative samples   (b) Increasing positive samples

Figure 3: Comparison between the average predictive accuracy of MIL (Metagol) vs Deep Learning (Siamese Net) on Task Classification dataset. (a) learning with one positive example and one or more negative examples. (b) learning with one or more positive examples, and the same number of negatives. The training and test data have been randomly chosen from the Task Classification dataset and the test data includes 15 positive and 14 negative examples.

As shown in Figure (3a), the Siamese model started from 47% average accuracy with one positive and one negative example and stays around the default accuracy after learning from 6 examples(50%). Our system achieved an average accuracy of 70% from one positive and one negative example and maintained higher average accuracy than the Siamese model throughout. Figure (3b), depicts the average accuracy of our system and Siamese model in the case of an increasing number of positive examples. It also confirms the outperformance of our model over Siamese model.

To evaluate our model on a publicly available dataset, we use News Category Dataset [9] which contains news articles from different sources and covers various topics such as sports, technology and the environment. We randomly selected five positive examples from the environment news headlines to train our model and test the gain hypotheses on randomly selected 50 positive and 50 negative samples. As shown in Figure 4, the results suggest the advantage of our method based on MIL over the Siamese model when we have few training examples, e.g. one positive and one negative example.

It is noticeable that for training and evaluating a large dataset, we use two techniques as follows:

1. To solve the problem of requiring large amounts of background knowledge for training and testing, we use the background knowledge of each example separately.

2. To train our model on a large dataset with only one or a few examples (one-shot learning), we randomly select 10 samples and make an average from several possible one-shots to calculate the average of different types of one-shots (ex: the average of possible one positive and one negative example shots or an average of one positive and two negative examples etc).

(a) Increasing negative samples
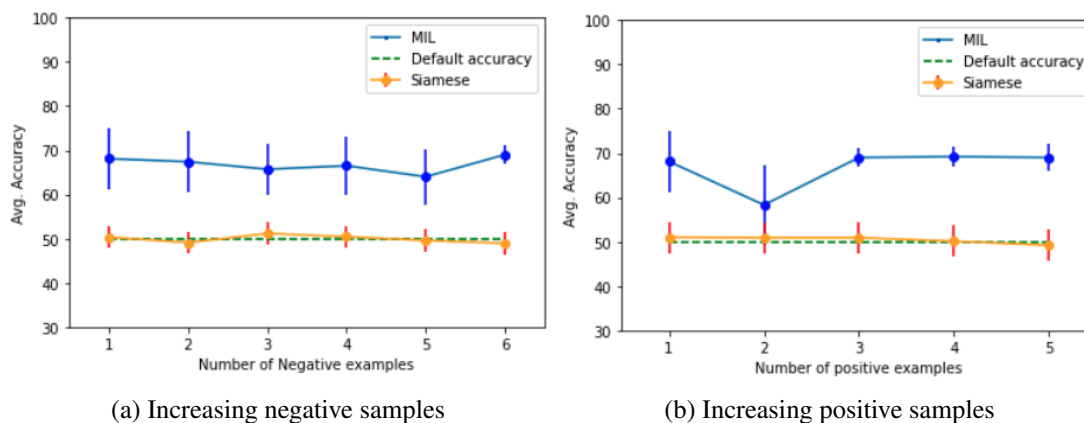
(b) Increasing positive samples

Figure 4: Comparison between the average predictive accuracy of MIL (Metagol) vs Deep Learning (Siamese Net) on News Category dataset. (a) learning with one positive example and one or more negative examples. (b) learning with one or more positive examples, and the same number of negatives. The training and test data have been randomly chosen from News Category Dataset and the test data includes 50 positive and 50 negative examples.

5   Conclusion

Short text classification is challenging due to the freely constructed sentence structures and their limited length. This study aims to provide a novel algorithm to classify short texts called "tasks" automatically based on the user's interests and without having any prior information about user behaviours. Thus, the first research question sought to determine a reliable source for providing pertinent background information to feed the system. We utilised Conceptnet to prepare the required background information related to the task. Our model was then designed based on different types of ILP and MIL systems, including Aleph and Metagol. We also used a Siamese Network model to compare with the deep learning methods. Siamese nets are the most popular DL approaches for one or few-shot learning. We applied the Convolutional-Siamese model with one positive and one or more negative examples. According to the current results, the MIL-based approach achieved 70% accuracy for one-shot learning, while the accuracy of ILP and Deep Learning was around default accuracy (50%) in this task.

Finally, we used News Category Dataset as a publicly available dataset to evaluate our model. This led us to utilise two techniques, including Background Knowledge Splitting and the Average One Shot Learning approach to train and evaluate the large dataset. The final results validated MIL's superior performance to the Siamese network for one-shot learning from text.

Supplemental Materials
The data and code used in the experiments can be found in the following Github repository:
https://github.com/ghazalmilani/One-Shot-Learning-from-Text-ICLP2023.git

References

[1] Davide Chicco (2021): *Siamese Neural Networks: An Overview*, pp. 73–94. Springer US, New York, NY, doi:10.1007/978-1-0716-0826-5_3.

[2] Andrew Cropper & Sebastijan Dumančić (2022): *Inductive logic programming at 30: a new introduction.* Journal of Artificial Intelligence Research 74, pp. 765–850, doi:10.1613/jair.1.13507.

[3] Andrew Cropper & Stephen H. Muggleton (2016): *Metagol System.* https://github.com/metagol/metagol. Available at `https://github.com/metagol/metagol`.

[4] Andrew Cropper & Sophie Tourret (2020): *Logical reduction of metarules.* Machine Learning 109, pp. 1323–1369, doi:10.1007/s10994-019-05834-x.

[5] Ji Han, Serhad Sarica, Feng Shi & Jianxi Luo (2020): *Semantic Networks for Engineering Design: A Survey.* arXiv:2012.07060.

[6] Filip Ilievski, Alessandro Oltramari, Kaixin Ma, Bin Zhang, Deborah L McGuinness & Pedro Szekely (2021): *Dimensions of commonsense knowledge.* Knowledge-Based Systems 229, p. 107347, doi:10.1016/j.knosys.2021.107347.

[7] Yann LeCun, Yoshua Bengio & Geoffrey Hinton (2015): *Deep learning.* nature 521(7553), pp. 436–444, doi:10.1038/nature14539.

[8] Dianhuan Lin, Eyal Dechter, Kevin Ellis, Joshua B Tenenbaum & Stephen H Muggleton (2014): *Bias reformulation for one-shot function induction.* doi:10.3233/978-1-61499-419-0-525.

[9] Rishabh Misra (2022): *News category dataset.* arXiv preprint arXiv:2209.11429, doi:10.48550/arXiv.2209.11429.

[10] Rolf Morel, Andrew Cropper & C.-H. Luke Ong (2019): *Typed Meta-interpretive Learning of Logic Programs.* In: JELIA, doi:10.1007/978-3-030-19570-0_13.

[11] Stephen Muggleton (2009): *Inverse entailment and progol.* New Generation Computing 13, pp. 245–286, doi:10.1007/BF03037227.

[12] Stephen Muggleton, Wang-Zhou Dai, Claude Sammut, Alireza Tamaddoni-Nezhad, Jing Wen & Zhi-Hua Zhou (2018): *Meta-interpretive learning from noisy images.* Machine Learning 107(7), pp. 1097–1118, doi:10.1007/s10994-018-5710-8.

[13] Stephen Muggleton & Luc de Raedt (1994): *Inductive Logic Programming: Theory and methods.* The Journal of Logic Programming 19-20, pp. 629–679, doi:10.1016/0743-1066(94)90035-3. Special Issue: Ten Years of Logic Programming.

[14] Stephen Muggleton, Dianhuan Lin, Niels Pahlavi & Alireza Tamaddoni-Nezhad (2014): *Meta-interpretive learning: Application to grammatical inference.* Machine Learning 94, doi:10.1007/s10994-013-5358-3.

[15] Stephen Muggleton, Dianhuan Lin & Alireza Tamaddoni-Nezhad (2015): *Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited.* Machine Learning 100, pp. 49–73, doi:10.1007/s10994-014-5471-y.

[16] Stephen H. Muggleton (2017): *Meta-Interpretive Learning: Achievements and Challenges (Invited Paper).* In Stefania Costantini, Enrico Franconi, William Van Woensel, Roman Kontchakov, Fariba Sadri & Dumitru Roman, editors: *Rules and Reasoning*, Springer International Publishing, Cham, pp. 1–6, doi:10.1007/978-3-319-61252-2_1.

[17] Stephen H Muggleton, Ute Schmid, Christina Zeller, Alireza Tamaddoni-Nezhad & Tarek Besold (2018): *Ultra-strong machine learning: comprehensibility of programs learned with ILP.* Machine Learning 107, pp. 1119–1140, doi:10.1007/s10994-018-5707-3.

[18] S.H. Nienhuys-Cheng, R. de Wolf, J.G. Carbonell & J. Siekmann (1997): *Foundations of Inductive Logic Programming.* Lecture Notes in Artificial Intelligence, Springer, doi:10.1007/3-540-62927-0.

[19] S. Patsantzis & S. Muggleton (2022): *Meta-interpretive learning as metarule specialisation.* Machine Learning, doi:10.1007/s10994-022-06156-1.

[20] SJ Russell & Peter Norvig (2022): *Artificial Intelligence: A Modern Approach, 4th, Global ed*.

[21] Robyn Speer, Joshua Chin & Catherine Havasi (2017): *Conceptnet 5.5: An open multilingual graph of general knowledge*. In: *Proceedings of the AAAI conference on artificial intelligence*, 31, doi:10.1609/aaai.v31i1.11164.

[22] A Srinivasan (2007): *Aleph: A learning engine for proposing hypotheses. Software available at http://web2. comlab. ox. ac. uk/oucl/research/areas/machlearn/Aleph/aleph. pl*. Available at `https://www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph.html`.

[23] Andrew Cropper Stephen Muggleton (2015): *Can predicate invention compensate for incomplete background knowledge?* In: *Thirteenth Scandinavian Conference on Artificial Intelligence: SCAI 2015*, 278, IOS Press, p. 27, doi:10.3233/978-1-61499-589-0-27.

[24] Alireza Tamaddoni-Nezhad, David Bohan, Ghazal Afroozi Milani, Alan Raybould & Stephen Muggleton (2021): *Human–Machine Scientific Discovery*, doi:10.1093/oso/9780198862536.003.0015.

[25] Dany Varghese, Roman Bauer, Daniel Baxter-Beard, Stephen Muggleton & Alireza Tamaddoni-Nezhad (2021): *Human-like rule learning from images using one-shot hypothesis derivation*. In: *International Conference on Inductive Logic Programming*, Springer, pp. 234–250, doi:10.1007/978-3-030-97454-1_17.

[26] Bin Wang, Guangtao Wang, Jing Huang, Jiaxuan You, Jure Leskovec & C-C Jay Kuo (2021): *Inductive learning on commonsense knowledge graph completion*. In: *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1–8, doi:10.1109/IJCNN52387.2021.9534355.

[27] Leiming Yan, Yuhui Zheng & Jie Cao (2018): *Few-Shot Learning for Short Text Classification*. Multimedia Tools Appl. 77(22), p. 29799–29810, doi:10.1007/s11042-018-5772-4.