Exploring Knowledge Extraction Techniques for System Dynamics Modelling: Comparative Analysis and Considerations

Roos M. Bakker[1,2][0000−0002−1760−2740], Gino J. Kalkman[1][0000−0001−8533−5384], Ioannis Tolios[1][0000−0002−0462−7156], Dominique Blok[1], Guido A. Veldhuis[1][0000−0003−1816−2641], Stephan Raaijmakers[1,2][0000−0003−2984−6889], and Maaike H.T. de Boer[1][0000−0002−2775−8351]

[1] Netherlands Organisation for Applied Scientific Research (TNO)
[2] University of Leiden
{roos.bakker, gino.kalkman, ioannis.tolios, dominique.blok, guido.veldhuis, stephan.raaijmakers, maaike.deboer}@tno.nl

Abstract. Knowledge graphs, semantic networks, and similar formal knowledge modelling structures have become popular solutions to express linguistic entities and relations between them. They are computer representations of human knowledge with all its semantic richness, and therefore powerful tools for structuring all kinds of data. Creating such models is, however, an extensive task which involves meticulous manual work by developers and experts. With the large amount of online textual information, models ideally can be automatically extracted, enriched, and maintained using Natural Language Processing (NLP) techniques. In this paper, we explore these techniques and present a comparative study of traditional knowledge graph extraction techniques (Keyword Extraction, Named Entity Recognition, co-occurrences) and a state-of-the-art large language model (LLM) in the context of systems dynamics modelling. System dynamics modelling is an approach to model the behaviour of complex systems, such as policy developments and politics. Often graphs and diagrams are used to support this process. The main contributions of this paper are twofold: first, a comparison and evaluation of traditional NLP methods compared to an LLM; second, an interface for including human feedback, enhancing future collaboration between system and domain expert. Our study informs practitioners about the suitability of these techniques to support problem analysis using system dynamics modelling and paves the way for future research in refining and integrating approaches.

Keywords: Information Extraction · Knowledge Graphs · LLM · System Dynamics

1 Introduction

With the current large amount of information, the demand for methods to structure and harness knowledge from textual sources has grown [30]. Knowledge

graphs, semantic networks, and similar models such as ontologies are used to answer this demand, they allow for better understanding, interoperability, and search-ability of our data [31]. Knowledge graphs have gained increasingly more attention in recent years from both academia and industry [34]. They represent both concepts and the relations between them in an intuitive manner. However, creating them is an extensive task which involves manual work from developers and domain experts.

Smaller, strict graphs such as ontologies especially require specialist knowledge to create and keep them up-to-date. One example of a field where such graphs are often used is system dynamics. This field aims at understanding complex real-life problems by modelling them; thereby defining the structure and behaviour of a problem [56]. System dynamics modellers can use all kinds of information, written, numerical, but they also depend heavily on expert knowledge. Recently, using text as a knowledge source has gained attention [57,24], but automatic extraction techniques have not been explored in this field. Techniques based on Natural Language Processing (NLP) such as knowledge graph extraction, completion, and ontology learning can support this process.

In this paper, we explore several Natural Language Processing (NLP) techniques for automatically extracting graphs. Our main goal is to study how we can automatically extract knowledge graphs of high quality that are usable for an analytical task such as system dynamics modelling. The main contributions of this paper are twofold: first, the comparison and evaluation of traditional NLP methods compared to an LLM; second, the introduction of a human feedback interface enhancing collaboration between system and domain expert.

In the next section, we will shortly summarise the field of knowledge graph extraction, ontology learning and the information extraction techniques used in this paper. In section 3, we present an architecture for graph extraction, the data and use case, and the algorithms and tools that we used. In section 4, we will present a qualitative analysis of the performance of our architecture and algorithms and introduce the user interface. Finally, we will conclude the paper with a short summary, a discussion of the results, and an outlook for future research.

2 Background

The field of ontology learning, and more recently knowledge graph extraction, underwent large changes with the rise of data-driven methods. Early approaches primarily relied on rule-based systems. Machine learning methods sprouted quickly since the 2010s, which lead to methods that could learn patterns and relations from text. A short overview and history of the field is given in section 2.1. Statistical methods also increased the size and nature of knowledge graphs and how they are used within applications. In section 2.2 we will discuss often used information extraction techniques for knowledge graphs, and the techniques used in this paper, such as LLMs, which will be discussed in section 2.3.

2.1 Ontology Learning

Ontology learning is the field that is focused on learning ontologies based on data [19] [11]. A well known concept in ontology learning is the ontology learning layer cake [13], consisting of layers for extractions for ontologies, increasing in complexity: extracting terms, synonyms, concept formation, concept hierarchy, relations, relation hierarchy, axiom schemata and finally, the extraction of general axioms. Wong et al. [61] give an overview of the relations and tasks related to ontology learning, such as part-of-speech tagging, dependency parsing and inductive logic programming. They mention that the techniques using in this field are categorised as statistics-based, linguistics-based, logic-based, or hybrid. Several ontology learning systems have been proposed such as ASIUM [26], GATE [10] and Text2Onto [14], as compared in [53]. In the early 2000s, solutions mostly built upon an existing ontology, only considered taxonomic relations, and no automatic ontology learning was involved. In the decade after this early work, the hybrid approach using linguistic and statistical techniques became the standard, as performance in the fields of Natural Language Processing and Machine Learning improved and more large (web) data became available to make this possible [4].

Currently, several ontology learning tools are available that handle several parts of the layered cake, of which most are published more than a decade ago [37]. Khadir et al. [37] mention that currently none of the methods are anywhere near producing a ready-to-use ontology. This is, according to the authors, due to lack of good training data for deep learning methods, the fact that these methods are language dependent and not universal, and a lack of good automatic evaluation. Human additions, feedback, and reviews are, therefore, still necessary. Several works propose tools or methods for facilitating this, such as [46] with a focus on annotation of text, and [7], where feedback is processed in a multi-agent system.

The field of ontology learning traditionally focused on ontologies. Since the introduction of machine learning implementations, the term knowledge graph has become popular, especially with the introduction of the Google Knowledge Graph in 2012 [54]. Soon, the industry used the term more and academia also picked up on the term around 2015 [38,27]. The term is mostly used to describe an ontology and its instances, that is, the schema and the data. Sometimes the difference between the ontology and the instances is made, and an ontology is used to add a layer of semantics to the terms in the knowledge graph [32]. With this development, ontology learning evolved into several subtopics such as knowledge graph extraction [9,39] and knowledge graph completion [17]. Simultaneously, the field of NLP changed dramatically, due to the advancements in machine learning.

2.2 Information Extraction for Knowledge Graphs

The field of Information Extraction has always been closely linked to the field of ontology learning and more recently knowledge graph extraction. One of the

fields that focuses on information extraction for knowledge graphs is named Open Information Extraction (OpenIE) [45]. Triples in the form of a subject, verb or relation, and object are created from large amounts of text. Methods include TextRunner, WOE (pos and parse), ReVerb, KrakeN, EXEMPLAR, OLLIE, PredPatt, ClausIE, OpenIE4, CSD-IE, NESTIE, MinIE, and Graphene among others [45,28].

These approaches use a variety of NLP techniques. For producing graphs, often multiple techniques are combined, where entities and relations are extracted and then linked in a different step [36]. Named Entity Recognition (NER) is an example of a technique which can function as a first step for knowledge graph extraction, even though it has not often been used yet [2]. NER involves identifying entities in texts, such as persons or locations. It often goes hand in hand with Named Entity Linking (NEL), where entities are linked in an existing graph or database [2]. The first approaches were rule-based, but now mostly statistical approaches are used. Keyword extraction underwent a similar development. Recent approaches such as YAKE [15] and BERT-based approaches [23] showed good performance, producing accurate keywords for its corresponding texts.

Besides the extraction of terms or concepts from text, the extraction of the relation between those concepts is essential for forming a graph. This subfield is named Relation Extraction, or Link Prediction. In the past decade, machine learning approaches were used to extract relations, using supervised approaches, semi-supervised approaches or bootstrapping approaches (using hand-crafted extraction patterns), often viewing the relation extraction as a binary classification problem [5]. More recently, deep learning methods are often used [34], named neural relation extraction. Several different types of neural networks are used, from CNNs to LSTMs, and recently also Graph Neural Networks, as they can represent the data in a knowledge graph-like manner. GNNs can also be used to joinly learn entities and relations. It is mentioned by [34] that often relation extraction suffers from noisy patterns. Attention mechanisms and reinforcement learning could reduce the noise, as well as capturing richer representations.

2.3 LLMs and knowledge representation

LLMs are the natural progression of stochastic NLP models like Hidden Markov Models, and, more recently, word embedding models like Word2Vec [42], BERT [23] and XLNET [63]. Current LLMs are typically produced by Transformers [59], with decoder-only models (e.g. GPT models [47], Palm [18], Llama [58]) becoming ever more dominant. A typical LLM has (hundreds of) billions of parameters: the neural weights on the connections between neurons in the various layers stacked up in the encoder/decoder blocks.

LLMs have been demonstrated to absorb, infer and reproduce factual information from self-supervised training on large quantities of texts [12]. This raises interesting questions from the perspective of knowledge representation: to what extent do these models go beyond memorizing their training data? Can LLMs infer structured knowledge from their textual training data, like semantic relationships, and derive logical entailment patterns or even new insights based

on this knowledge, rather than just echoing observed training data? How would such knowledge be represented in - and be observable from- the parametric, distributed underlying neural architectures? This topic is still in its infancy. In [60], various emergent properties of LLMs have been demonstrated to occur beyond certain scales (including semantic entailment and reasoning) but knowledge inference and representation was not part of that inventory. OpenAI has released tooling for qualitative analysis of neuron activity in GPT models [8] that goes some way to answering these questions by analyzing which specific textual inputs trigger certain neurons in GPT models.

While LLMs possess powerful memories that typically are trained in just a few epochs, the factual information they contain is based on the static snapshots of their training data. LLMs have been criticised for low adherence to such factual information [49]: due to (a.o.) the well-known LLM hallucination problem [35], factual consistency is not guaranteed for LLM-based text production, with hallucinations ranging from lexical substitutions to illicit reasoning patterns. The poor alignment of LLMs with factual data is addressed with through so-called memory augmentation techniques that attempt to infuse external data sources into the LLM architecture (e.g. [50]).

A significant body of research addresses the problem of carving out stored information from LLMs directly. Popular approaches consist of developing prompting schemes ("in-context learning", [44]) that poll LLMs for completion of incomplete patterns. In [16], this technique is shown to reconstruct verbatim training data from LLMs. Similarly, recent work polls LLMs through manual prompting for completing subject-predicate-object triples (link prediction; [3]). Such approaches lead to new divisions of labor between LLMs and knowledge graphs: LLMs serve to infer or complete knowledge graphs rather than producing these graphs directly (see e.g. [48]).

3 Method

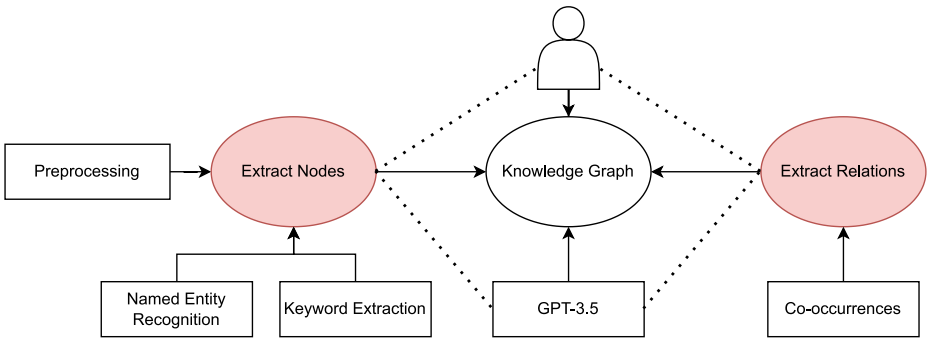

Fig. 1. Overview and architecture of method

In this paper, we do a comparative exploration of some traditional NLP methods and state-of-the-art approaches for knowledge graph extraction. We apply these techniques on a use case from the system dynamics field, which is described in section 3.1. Different NLP methods are combined to extract graphs. An architecture is shown in figure 1. First, we extract nodes for the graph using keyword extraction and NER. Second, we extract edges exploring techniques such as co-occurrences. These approaches can be combined in a graph. More details are given in section 3.2. Recent work on LLMs has shown its potential on combining these steps as discussed in section 2.3. Section 3.3 describes our method for extracting KGs using a well-known LLM: GPT-3.5 [12]. Finally, an interface was designed to include human feedback in the output graphs. The methods for the interface are shortly described in section 3.4. We evaluated the results of each approach qualitatively. The first two and the last author of this paper reviewed the results individually. We evaluated the results on different aspects: the amount of high quality keywords or relations, the amount of noise such as stopwords, the overlap between different n-grams, and the total amount of the results (where applicable). Additionally two domain experts were asked to give a general judgement on the results of each step.

3.1 Data and Use Case

A system dynamics modeller seeks to provide insight into the origin of system behaviour based on the causal structure underlying the behaviour. The causal structure consists of variables and relations. These variables express the stocks, flows, information and decisions that govern the behaviour of the system. The system and behaviour can be as small as processes in the human body or as large as the strategic behaviour of leaders in the geopolitical arena. In this paper we will study an example of the latter. Text documents often form an important source for model development. In this example, we will consider a modeller who seeks to understand the geopolitical behaviour and relations of the Russian Federation. A modeller might for instance be interested in including variables that explain why the relation between the Russian Federation and the western world shifted from cooperation to confrontation. In this paper we will test and compare the performance of the algorithms by applying it on the paper of McFaul from 2020 [41]. In this paper, the influence of President Putin and domestic factors on Russian foreign policy are described. It provides valuable information for a system dynamics modeller, such as a description of historic behaviour and associated variables and causal relations. It is also a very dense paper which covers many different world leaders, nations, Russian institutions and the factors that influence their behaviour. An algorithm that supports the modeller to quickly organise this information would benefit the modelling process. The paper is written in English and contains 20.508 words. The paper does not contain information other than text.

3.2 Traditional NLP methods

Node Extraction As mentioned in section 2.1, the nodes of a knowledge graph represent the central concepts in a particular knowledge domain. One method that can be used to identify these concepts in a text is that of keyword extraction, which facilitates the identification of terms and phrases that are most relevant to a given text. In this work, we used the keyword extraction approach as it is implemented in KeyBERT [29].

KeyBERT uses a transformer model to translate a document into an embedding, a vector describing the semantics of that particular document. Subsequently, the same model is used to generate embeddings for each word and phrase in that document. The algorithm then calculates similarity scores between each of the word/phrase embeddings on the one hand and the document embedding on the other hand. These similarity scores are used to identify the words and phrases that are most similar to the document as a whole. These words and phrases are considered to be the document's key concepts. Entities can also be extracted from text using Named Entity Recognition (NER), as described in section 2.2. To extract named entities, we utilised the Spacy NER functionality and employed its large English model [33]. In the NER method, we filter on labels with the tag "PERSON", "ORG", "GPE" and a link to Wikipedia. In the Spacy method we do not filter the results on labels. Additionally, we compare NER and Spacy with Yake (Yet Another Keyword Extractor) [15] and Rake (Rapid Automatic Keyword Extraction) [52][3].

Relation Extraction As discussed in section 2.2, there are several methods towards extracting relations from text. In this work, we implemented the co-occurrence algorithm. Co-occurrence analysis extracts relations dsbased on the number of times word pairs appear together, often in the same sentence [21,40]. The algorithm only takes the concepts into consideration, so the relation between them remains generic. For more specific relations, this approach could be combined with other approaches as described in section 2.2.

To identify the word pairs, we computed sets of distinct word pairs that co-occur within sentences from the data described in section 3.1. We used a maximum word distance of 5, so the word pair must occur withing 5 words of each other. For this process, we used the N-gram generator within the CountVectorizer module of the Scikit-learn package [55]. We set a threshold of 0.5 to limit the amount of word pairs to only those that occur a minimum number of times.

3.3 LLMs

To compare the methods described above with a state-of-the-art LLM, we queried the GPT-3.5 model with sentences or paragraphs from the system dynamics use case described in section 3.1. Due to maximum token restriction, we limited our

[3] For access to the code, data, or any related materials used in this work, please contact the authors.

experiments to the first page of the paper, which introduces and summarises the topic. The best prompt was found with prompt engineering. We experimented on three different aspects: sentence level versus paragraph level, with versus without added keywords, and on output format. The output format can influence the richness of the results, when the model can successfully produce RDF or a similar format the graph can contain a richer set of relations and axioms. For measuring performance with or without keywords, we use the extracted keywords with the methods described above on a document level. The different combinations of settings and example prompts are shown in table 1. Since this is an information extraction task, we set the temperature relatively low on 0.3. This reduces the creativity of the model, leading to more certainty in its answers, which is recommended since we are asking for a single output [62]. The other parameters were left at default settings.

Exp.	Keywords	Level	Output	Prompt
1	yes	Sentence	Triple	Take the following sentence: {sentence}. Extract all relations in this sentence between any of the following keywords: {keywords}. Use this format: <entity1>- <entity2>- <relation class>.
2	yes	Paragraph	Triple	Prompt of experiment 1 with sentence replaced by paragraph
3	yes	Sentence	RDF	Prompt of experiment 1 with the format replaced by "RDF"
4	yes	Paragraph	RDF	Prompt of experiment 2 with the format replaced by "RDF"
5	no	Sentence	Triple	Prompt of experiment 1 with the keywords left out
6	no	Paragraph	Triple	Prompt of experiment 2 with the keywords left out
7	no	Sentence	RDF	Prompt of experiment 3 with the keywords left out
8	no	Paragraph	RDF	Prompt of experiment 4 with the keywords left out

Table 1. All combinations of prompt experiments

3.4 Interface

A user interface was designed and developed to assist domain experts in evaluating generated knowledge graphs. It is an application, built using web-based technologies. The purpose of this tool is twofold. On one hand, accelerate the knowledge graph generation process by automatically creating a first version of a knowledge graph about a domain, based on a set of relevant documents. On the other hand, the domain experts can provide their feedback to the generated knowledge graph. The node and relation extraction methods from section 3.2 are integrated as back-end.

In our application, an ontology is generated in Turtle format. The concepts of the ontology are expressed as owl:Classes, the relations between them as owl:ObjectProperties [6], while the added evaluation comments are expressed as

skos:editorialNotes [43]. The application was built using Vue.js and Vuetify [22]. D3.js [20] was used for the interactive visualisations, while RDFLib.js [51] for storing, parsing, serializing data into various formats, and keeping track of the additions in the ontology.

4 Results

4.1 Traditional NLP methods

Node Extraction For the qualitative analysis of the node extraction methods, we provide an analysis on sentence level and on document level. On sentence level, the results for an example sentence are shown in table 2. We observed that NER extracts dates, organisations and persons, but also cardinals (numbers), events and money. From the keyword-based methods, we observe that Spacy is an elaboration of the NER results; it outputs Named Entities, such as "Russia", extended with n-grams. Yake outputs many n-grams, such as "Russia move from cooperative" and "move from cooperative ties". Rake produces less results compared to Yake, but more noise than keyBERT, such as "u" and "generally".

Method	Result
NER	the United States (GPE), Russia (GPE), a few decades ago (DATE), U.S.- Russian (NORP), West (LOC)
Spacy	a few decades ago, U.S.-Russian, Russia, the United States, West
keyBert	russia, strategic partnerships, cooperative ties, relations, conflict, relationship, international norms, domestic goals, united states, west, few decades, new era
Yake	United States and Russia move, Russia move from cooperative ties, United States and Russia, Russian relations and Russia, States and Russia move, shared domestic goals, Russia move from cooperative, move from cooperative ties, relations and Russia s relationship, norms a few decades ago, international norms a few decades, United States, States and Russia, West more generally, Russian relations, strategic partnerships, Russia move, relations and Russia, cooperative ties, shared domestic, domestic goals, move from cooperative, international norms, decades ago, era of conflict
Rake	shared domestic goals, russian relations, united states, strategic partnerships, new era, international norms, decades ago, cooperative ties, russia move, russia, west, u, relationship, generally, conflict

Table 2. Result for example sentence: *How did the United States and Russia move from cooperative ties, strategic partnerships, shared domestic goals, and international norms a few decades ago to a new era of conflict in U.S.-Russian relations and Russia s relationship with the West more generally?*

On document level, NER provides 370 results, while for the keyword-based methods we can choose the number of wanted results. For the keyword-based

models (top 25), we observe that the Spacy model only outputs 1-grams, of which most of them are Named Entities, such as “Libya”, “US” and “Iranian”. Rake outputs n-grams, but these seem not of consistent quality, for example in “donald trump jr.”, “campaign chairman paul manafort”. Yake on the other hand outputs more consistent phrases, but not always relevant ones, such as “states or the ideological” and “Syria as well as economic”. KeyBERT outputs mostly 2- and 3-grams, of which several overlap, such as “russian foreign policy” and “russian foreign policy behaviour”. With a higher diversity parameter (0.7), these overlaps disappear. Example keywords are “cuban missile crisis”, “president boris yeltsin”, “russian domestic drama” and “subsequent policy responses”.

Interpreting these results, we can conclude that NER can provide good results when looking for a model with only Named Entities. Keywords provide a richer result, but maybe also contain more noise. From the keyword-based methods KeyBERT seems to provide a good combination of not only 1-grams, and relevant results, especially when working with the parameters such as diversity. When facing a larger document, the NER based approaches provide many results, whereas the keyword-based approaches can be set to provide a fixed number. Another approach could be to first select key sentences and then applying NER or keyword-based approaches on these sentences.

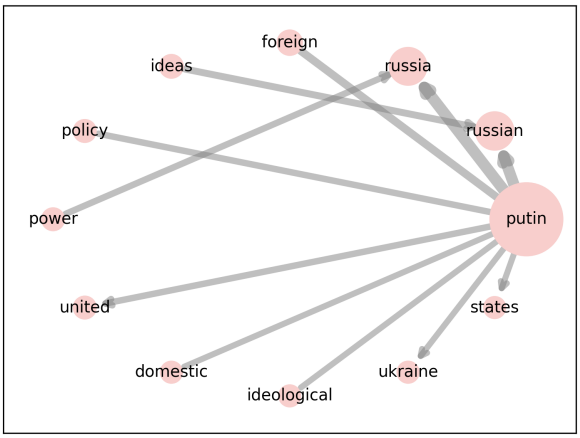

Fig. 2. A visualisation of results from the co-occurrence algorithm

Relation Extraction As described in section 3.2, we implemented a co-occurrence algorithm and applied it on the use case. The resulting graph is visualised in figure 2. Often occurring relations and nodes are bigger than less occurring ones.

The algorithm only recognised 1-grams, leading to terms such as “united” and “states” being separated. The concepts “Putin” and “Russia” occur together 23 times, which suggests a strong association in the text between them. Since the use case is an analysis about Putin, Russia, and their politics, this outcome is to be expected. Other often occurring terms are “policy” and “Putin”, “power” and “Russia”, and “Putin” and “Ukraine”. This graph does give a general idea about the content of the use case text, albeit on a high level.

4.2 LLMs

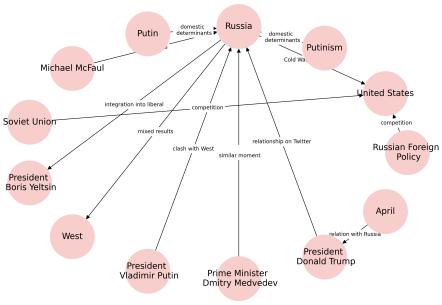

Fig. 3. Experiment 2

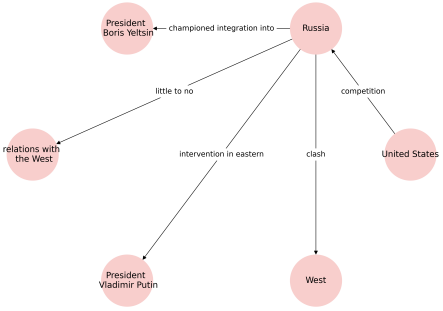

Fig. 4. Experiment 6

We queried the GPT-3.5 model with the settings and prompts described in section 3.3. Key observations are summarised in table 3. For the experiments using RDF output, it becomes clear that GPT cannot handle this request consistently. We experimented with different formats, such as asking for TTL or json-ld, but responses varied between at worse “I’m sorry, but I’m unable to generate a Turtle (TTL) format for the given sentence.”, and at best a large variation on notations. When asking for a simplified triple notation <entity1> - <entity2> - <relation>, GPT mostly complies, returning only few non-triple outputs.

Creating a graph based on sentence level or on paragraph level leads to differences in the detail of the resulting graph. Querying on sentence level leads to more nodes and relations, but also introduces noise due to not every sentence being equally important. For example, in our use case, a sentence occurs where the author quotes a certain president: “I am sometimes confused: is this 2016 or 1962?” [41]. This sentence results in triples such as “this - 2016 - is”, “I - 1962 - relation class: temporal”, for both experiment 1 and 5. When looking at paragraph level, less nodes and relations are extracted, but all the nodes are informative terms; no articles or personal pronouns are included. A visualisation of experiment 2 to illustrate this is shown in figure 3.

Finally, when looking at the influence of keywords on a query, it becomes clear that adding keywords influences the number of triples that are returned.

This difference can clearly be seen between experiments 1 and 2, and 5 and 6. Unfortunately, experiments 3, 4, 7, and 8 varied too much in output to see how keywords influenced the results. Comparing figure 4 and figure 3, the addition of keywords lead to more concepts, but also introduced some less logical concepts and relations such as "Russia - West - mixed result". Ultimately, whether to include keywords or not depends in the preferences of the user and the use case. For a user who rather have more results with less accuracy, including keywords and even running the model on sentences might be preferable.

Exp.	Keywords	Level	Output	Observations and examples
1	yes	Sentence	Triple	Many nodes (67), 2 non-triple outputs, non-concept nodes such as *this*, e.g. *Putin - Dometic Determinants - influences*
2	yes	Paragraph	Triple	Less nodes (20), high quality nodes, e.g. *Russian Foreign Policy - United States - competition*
3	yes	Sentence	RDF	Differences in output, e.g. *<subject3> = Putinism, <relation3> = hasDeterminant, <object3> = Domestic Determinants; 1. Relation: collapse, Subject: Soviet Union, Object: None*
4	yes	Paragraph	RDF	RDF output containing only dc:relations, e.g. *<dc:relation> Putinism <dc:relation> <dc:relation> Russian Foreign Policy <dc:relation>*
5	no	Sentence	Triple	Less nodes (35), 5 non-triple outputs, low quality nodes e.g. *U.S.-Russian confrontation - today - as high as*
6	no	Paragraph	Triple	Small amount of nodes (9), high quality nodes e.g. *United States - Russia - competition*
7	no	Sentence	RDF	RDF output similar to exp. 3
8	no	Paragraph	RDF	after multiple runs output in consistent format, e.g. *:Putin a :Person; :Putinism a :Concept*

Table 3. Key observations per experiment for GPT-3.5 on graph extraction

4.3 Interface

To accommodate user feedback on resulting graphs, we introduce a user interface. Figure 5 shows the interface for extracting a graph from text and providing feedback on the results, a larger version of the user interface is given in appendix A. The intended user is a domain expert. The interface consists of two main panes. On the left (pane A), the user can upload documents of interest or an existing ontology, choose an extraction algorithm and different values for its parameters. On the right (pane B), the generated ontology is visualised. A domain expert can inspect (search concepts of interest, get information about concepts) and interact (zooming, panning) with the generated ontology. The

user is able to obtain information for each selected concept and relation via the included information panel.

Users can assess the generated ontology by providing feedback in the form of comments via the information panel. They can provide feedback to the individual concepts, their in-between relations and the ontology as a whole. A concept can be characterised as "excellent", "okay" or "poor" (visually encoded by the color of each concept). For each concept evaluation, a justification is required. The evaluation of the relations and the ontology as a whole is more straightforward, since the user has to provide a comment. Whether a relation has been evaluated or not is visually encoded by the line pattern used for each edge. A straight line denotes a non-evaluated edge, while a dashed line denotes an evaluated one.

Currently, the interface has only been used by the authors for testing[4], but for future work it provides an opportunity to improve both the quality of the extracted graphs and the evaluation of them, as we will explain in section 5.

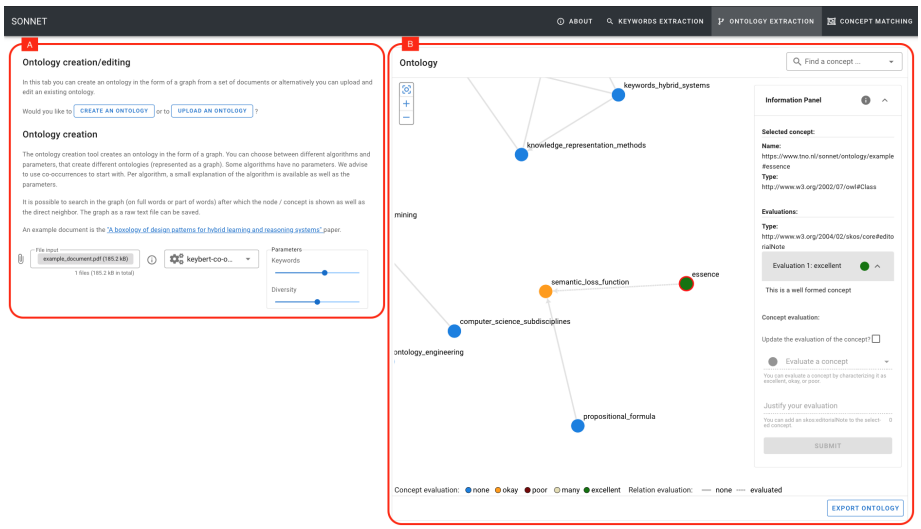

Fig. 5. The user interface for providing feedback on generated graphs

5 Conclusion & Discussion

In this paper, we explored the use of knowledge graph extraction techniques for a system dynamics use case. Knowledge graphs are a powerful solution to express entities and relations between them, but creating them involves manual work and time. Natural Language Processing (NLP) techniques can support this process. We compared some traditional NLP algorithms to a cutting-edge

[4] Access to the interface can be requested from the authors.

Large Language Model (LLM) in the context of systems dynamics modelling, and introduced an interface to allow for human feedback on the results. The results were qualitatively evaluated on their performance for the use case. They show that traditional techniques excel in extracting high quality domain concepts, while the LLM demonstrates adaptability and contextual understanding by recognising relations between domain concepts.

This paper primarily concentrates on qualitative analysis of the compared techniques within the context of a system dynamics modelling use case. For the field of system dynamics, an extensive ground truth for this task is hard to find due to the amount of manual expert work required. If such a ground truth were created, a quantitative analysis of a broader set of use cases could provide more insight on the performance on a larger scale. This work provides a first step on extracting a graph without such annotations. For a system dynamics modeler, this would be helpful in the starting phase of modelling a new domain. As the results show, high quality concepts and relations can be extracted, but the necessary nuances and complex causalities are not present yet. The proposed interface from the previous section can aid in adding feedback from the users, which both improves the results and gives insight in the quality. For instance, the amount of actions a user needs to make the graph of sufficient quality would be interesting to record. Differences between approaches can then give more insight in the performance, in addition to the qualitative analysis described above.

Results might also be improved by exploring more and improving on techniques. We only used a small set of information extraction techniques. There are more options available, and also within techniques, quality differs depending on the model or package used in implementation. For instance, an analysis of different keyword extraction and ontology extraction techniques is given in [9].

The performance of the LLM, in this case GPT-3.5, can be improved in several ways. Recently, OpenAI introduced the function calling functionality [25]. This assures the output adheres to a json format, which would solve problems with inconsistent outputs and makes quantitative evaluation possible. Due to time constraints, this option was not explored in this paper. Other promising approaches are few-shot learning instead of zero-shot learning [1], although creating examples for this task is challenging due to token limitations.

Finally, further improvements could be achieved by combining different techniques in a pipeline for this specific use case, where different algorithms might be used depending on the text. For instance, sentences could be sorted for relevance, or classified on whether they contain causal relations, which are especially important for a system dynamics modeler. A first version of a graph could then be produced with an LLM or co-occurrences. Next, human feedback indicates that some nodes are not relevant for the domain. These nodes could be replaced with similar extracted keywords. Human feedback could be used in other ways, ranging from interpreting the results in this work to creating a hybrid system where the system becomes adaptive to its users.

The main contributions of this paper are twofold: first, we compared and evaluated traditional NLP methods to an LLM; second, we introduced an in-

terface for including human feedback, enhancing collaboration between system and domain expert. Our study underscores the potential of NLP techniques in constructing knowledge graphs for system dynamic modelling, even though both traditional NLP approaches and recent LLMs have their merits and limitations. For future work, it would be interesting to explore and implement the points discussed above. Additionally, expanding the human feedback functionality would be an important tool for system dynamic modelling. For other fields and use cases, techniques should be tailored since different use cases require different levels of specificity. It would also be an interesting direction to link extracted graphs to an upper ontology, enabling advanced reasoning capabilities and a dynamic knowledge base for even more robust modelling and analysis.

Acknowledgements We would like to thank the internal Risicodragend Verkennend Onderzoek (RVO) project named Causal Relations for their financial support, as well as Rob van Waas for his project leading skills, insight in the data and constructive feedback.

References

1. Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., Sontag, D.: Large language models are few-shot clinical information extractors. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 1998–2022 (2022)
2. Al-Moslmi, T., Ocaña, M.G., Opdahl, A.L., Veres, C.: Named entity extraction for knowledge graphs: A literature overview. IEEE Access **8**, 32862–32881 (2020)
3. Alivanistos, D., Santamaría, S.B., Cochez, M., Kalo, J.C., van Krieken, E., Thanapalasingam, T.: Prompting as probing: Using language models for knowledge base construction (2023)
4. Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M.: A survey of ontology learning techniques and applications. Database **2018**, bay101 (2018)
5. Bach, N., Badaskar, S.: A review of relation extraction. Literature review for Language and Statistics II **2**, 1–15 (2007)
6. Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., et al.: Owl web ontology language reference. W3C recommendation **10**(2), 1–53 (2004)
7. Benomrane, S., Sellami, Z., Ayed, M.B.: An ontologist feedback driven ontology evolution with an adaptive multi-agent system. Advanced Engineering Informatics **30**(3), 337–353 (2016)
8. Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., Saunders, W.: Language models can explain neurons in language models — openaipublic.blob.core.windows.net. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html` (2023), [Accessed 24-08-2023]
9. de Boer, M., Verhoosel, J.P.: Towards data-driven ontologies: a filtering approach using keywords and natural language constructs. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. pp. 2285–2292 (2020)
10. Bontcheva, K., Tablan, V., Maynard, D., Cunningham, H.: Evolving gate to meet new challenges in language engineering. Natural Language Engineering **10**(3-4), 349–373 (2004)

11. Brewster, C.A.: Mind the gap: Bridging from text to ontological knowledge. Ph.D. thesis, University of Sheffield (2008)
12. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
13. Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: methods, evaluation and applications, vol. 123. IOS press (2005)
14. Buitelaar, P., Olejnik, D., Sintek, M.: A protégé plug-in for ontology extraction from text based on linguistic analysis. In: European Semantic Web Symposium. pp. 31–44. Springer (2004)
15. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A.: Yake! keyword extraction from single documents using multiple local features. Information Sciences **509**, 257–289 (2020)
16. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., Raffel, C.: Extracting training data from large language models (2021)
17. Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., Duan, Z.: Knowledge graph completion: A review. Ieee Access **8**, 192435–192456 (2020)
18. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
19. Cimiano, P., Mädche, A., Staab, S., Völker, J.: Ontology learning. In: Handbook on ontologies, pp. 245–267. Springer (2009)
20. D3.js developers: D3.js – data-driven documents (2009), `https://d3js.org/`, accessed on 30-08-2023
21. De Boer, M., Verhoosel, J.: Creating and evaluating data-driven ontologies. to appear (2019)
22. developers, V.: Vue.js – the progressive javascript framework v3.0 (2014), `https://vuejs.org/guide/introduction.html`, accessed on 30-08-2023
23. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
24. Eker, S., Zimmermann, N.: Using textual data in system dynamics model conceptualization. Systems **4**(3), 28 (2016)
25. Eleti, Atty and Harris, Jeff and Kilpatrick, Logan: Function calling and other API updates (2023), `https://openai.com/blog/function-calling-and-other-api-updates`, accessed: 15 August 2023
26. Faure, D., Nédellec, C.: Asium: Learning subcategorization frames and restrictions of selection. In: ECML98, Workshop on Text Mining. vol. 409, p. 410 (1998)
27. Fensel, D., Şimşek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J., Wahler, A., Fensel, D., et al.: Introduction: what is a knowledge graph? Knowledge graphs: Methodology, tools and selected use cases pp. 1–10 (2020)
28. Glauber, R., Claro, D.B.: A systematic mapping study on open information extraction. Expert Systems with Applications **112**, 372–387 (2018)
29. Grootendorst, M.: Keyword extraction with bert (2020), `https://towardsdatascience.com/keyword-extraction-with-bert-724efca412ea`, accessed on: 23-08-2023
30. Gutiérrez, C., Sequeda, J.F.: Knowledge graphs. Communications of the ACM **64**(3), 96–104 (2021)

31. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. ACM Computing Surveys (Csur) **54**(4), 1–37 (2021)
32. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. ACM Computing Surveys (Csur) **54**(4), 1–37 (2021)
33. Honnibal, M., Montani, I.: spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. GitHub repository (2017), `https://github.com/explosion/spaCy`
34. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE transactions on neural networks and learning systems **33.2**, 494–514 (2021)
35. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12) (mar 2023). `https://doi.org/10.1145/3571730`, `https://doi.org/10.1145/3571730`
36. Kertkeidkachorn, N., Ichise, R.: T2kg: An end-to-end system for creating knowledge graph from unstructured text. In: Workshops at the Thirty-First AAAI Conference on Artificial Intelligence (2017)
37. Khadir, A.C., Aliane, H., Guessoum, A.: Ontology learning: Grand tour and challenges. Computer Science Review **39**, 100339 (2021)
38. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29.1 (2015)
39. Martinez-Rodriguez, J.L., López-Arévalo, I., Rios-Alvarado, A.B.: Openie-based approach for knowledge graph construction from text. Expert Systems with Applications **113**, 339–355 (2018)
40. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools **13**(01), 157–169 (2004)
41. McFaul, M.: Putin, putinism, and the domestic determinants of russian foreign policy. International Security **45**(2), 95–139 (2020)
42. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR **abs/1310.4546** (2013), `http://arxiv.org/abs/1310.4546`
43. Miles, A., Pérez-Agüera, J.R.: Skos: Simple knowledge organisation for the web. Cataloging & Classification Quarterly **43**(3-4), 69–83 (2007)
44. Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., Zettlemoyer, L.: Rethinking the role of demonstrations: What makes in-context learning work? In: EMNLP (2022)
45. Niklaus, C., Cetto, M., Freitas, A., Handschuh, S.: A survey on open information extraction. arXiv preprint arXiv:1806.05599 (2018)
46. Opasjumruskit, K., Böning, S., Schindler, S., Peters, D.: Ontohuman: Ontology-based information extraction tools with human-in-the-loop interaction. In: International Conference on Cooperative Design, Visualization and Engineering. pp. 68–74. Springer (2022)
47. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback (2022)

48. Pan, J.Z., Razniewski, S., Kalo, J.C., Singhania, S., Chen, J., Dietze, S., Jabeen, H., Omeliyanenko, J., Zhang, W., Lissandrini, M., Biswas, R., de Melo, G., Bonifati, A., Vakaj, E., Dragoni, M., Graux, D.: Large language models and knowledge graphs: Opportunities and challenges (2023)
49. Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., Huang, Q., Liden, L., Yu, Z., Chen, W., Gao, J.: Check your facts and try again: Improving large language models with external knowledge and automated feedback (2023)
50. Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., Shoham, Y.: In-context retrieval-augmented language models (2023)
51. rdflib.js developers: rdflib.js – a javascript library for working with rdf (2011), https://github.com/linkeddata/rdflib.js/, accessed on 30-08-2023
52. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. Text mining: applications and theory pp. 1–20 (2010)
53. Shamsfard, M., Barforoush, A.A.: The state of the art in ontology learning: a framework for comparison. The Knowledge Engineering Review **18**(4), 293–316 (2003)
54. Singhal, A.: Introducing the knowledge graph: things, not strings, https://blog.google/products/search/introducing-knowledge-graph-things-not/
55. sklearn: Countvectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html, accessed: 2023-08-21
56. Sterman, J.D.: System dynamics modeling: tools for learning in a complex world. California management review **43**(4), 8–25 (2001)
57. Tomoaia-Cotisel, A., Allen, S.D., Kim, H., Andersen, D., Chalabi, Z.: Rigorously interpreted quotation analysis for evaluating causal loop diagrams in late-stage conceptualization. System Dynamics Review **38**(1), 41–80 (2022)
58. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
59. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5998–6008 (2017), https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
60. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E.H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., Fedus, W.: Emergent abilities of large language models (2022)
61. Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: A look back and into the future. ACM computing surveys (CSUR) **44**(4), 1–36 (2012)
62. Xu, F.F., Alon, U., Neubig, G., Hellendoorn, V.J.: A systematic evaluation of large language models of code. In: Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming. pp. 1–10 (2022)
63. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. CoRR **abs/1906.08237** (2019), http://arxiv.org/abs/1906.08237

Appendix A: Interface