Common-Knowledge Concept Recognition for SEVA

Jitin Krishnan[1], Patrick Coronado[2], Hemant Purohit[3], Huzefa Rangwala[4]
[1,4]Department of Computer Science, George Mason University
[2]Instrument Development Center, NASA Goddard Space Flight Center
[3]Information Sciences & Technology Department, George Mason University
jkrishn2@gmu.edu, patrick.l.coronado@nasa.gov, hpurohit@gmu.edu, rangwala@gmu.edu

Abstract

We build a common-knowledge concept recognition system for a Systems Engineer's Virtual Assistant (SEVA) which can be used for downstream tasks such as relation extraction, knowledge graph construction, and question-answering. The problem is formulated as a token classification task similar to named entity extraction. With the help of a domain expert and text processing methods, we construct a dataset annotated at the word-level by carefully defining a labelling scheme to train a sequence model to recognize systems engineering concepts. We use a pre-trained language model and fine-tune it with the labeled dataset of concepts. In addition, we also create some essential datasets for information such as abbreviations and definitions from the systems engineering domain. Finally, we construct a simple knowledge graph using these extracted concepts along with some hyponym relations.

Keywords: Natural Language Processing, Named Entity Recognition, Concept Recognition, Relation Extraction, Systems Engineering.

INTRODUCTION

The Systems Engineer's Virtual Assistant (SEVA) (Krishnan, Coronado, and Reed 2019) was introduced with the goal to assist systems engineers (SE) in their problem-solving abilities by keeping track of large amounts of information of a NASA-specific project and using the information to answer queries from the user. In this work, we address a system element by constructing a common-knowledge concept recognition system for improving the performance of SEVA, using the static knowledge collected from the Systems Engineering Handbook (NASA 2017) that is widely used in projects across the organization as domain-specific commonsense knowledge. At NASA, although there exists knowledge engines and ontologies for the SE domain such as MBSE (Hart 2015), IMCE (JPL 2016), and OpenCaesar (Elaasar 2019), generic commonsense acquisition is rarely discussed; we aim to address this challenge.

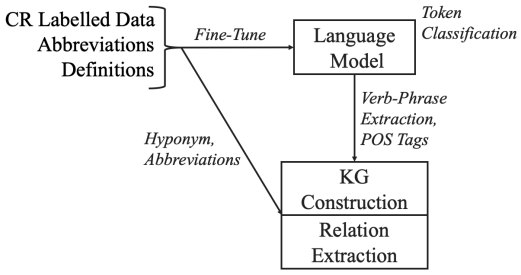


Figure 1: Common-knowledge concept recognition and simple relation extraction

SE commonsense comes from years of experience and learning which involves background knowledge that goes beyond any handbook. Although constructing an assistant like SEVA system is the overarching objective, a key problem to first address is to extract elementary common-knowledge concepts using the SE handbook and domain experts. We use the term *'common-knowledge'* as the *'commonsense'* knowledge of a specific domain. This knowledge can be seen as a pivot that can be used later to collect *'commonsense'* knowledge for the SE domain. We propose a preliminary research study that can pave a path towards a comprehensive commonsense knowledge acquisition for an effective Artificial Intelligence (AI) application for the SE domain. Overall structure of this work is summarized in Figure 1. Implementation with demo and dataset is available at: https://github.com/jitinkrishnan/NASA-SE .

BACKGROUND AND MOTIVATION

Creating commonsense AI still remains an important and challenging task in AI research today. Some of the inspiring works are the CYC project (Panton et al. 2006) that tries to serve as a foundational knowledge to all systems with millions of everyday life commonsense assertions, Mosaic Commonsense Knowledge Graphs and Reasoning (Zellers et al. 2018) that addresses aspects like social situations, mental states, and causal relationships, and Aristo System (AI2 Allen Institute for AI) that focuses on basic science knowledge. In NASA's context, systems engineering combines several engineering disciplines requiring extreme coordination and is prone to human errors. This, in combination with

the lack of efficient knowledge transfer of generic lessons-learned makes most technology-based missions risk-averse. Thus, a comprehensive commonsense engine can significantly enhance the productivity of any mission by letting the experts focus on what they do best.

Concept Recognition (CR) is a task identical to the traditional Named Entity Recognition (NER) problem. A typical NER task seeks to identify entities like name of a person such as *'Shakespeare'*, a geographical location such as *'London'*, or name of an organisation such as *'NASA'* from unstructured text. A supervised NER dataset consists of the above mentioned entities annotated at the word-token level using labelling schemes such as BIO which provides beginning (B), continuation or inside (I), and outside (O) representation for each word of an entity. (Baevski et al. 2019) is the current top-performing NER model for CoNLL-2003 shared task (Sang and De Meulder 2003). Off-the-shelf named entity extractors do not suffice in the SE common-knowledge scenario because the entities we want to extract are domain-specific concepts such as *'system architecture'* or *'functional requirements'* rather than physical entities such as *'Shakespeare'* or *'London'*. This requires defining new labels and fine-tuning.

Relation extraction tasks extract semantic relationships from text. These extractors aim to connect named entities such as *'Shakespeare'* and *'England'* using relations such as *'born-in'*. Relations can be as simple as using hand-built patterns or as challenging as using unsupervised methods like Open IE (Etzioni et al. 2011); with bootstrapping, supervised, and semi-supervised methods in between. (Xu and Barbosa 2019) and (Soares et al. 2019) are some of the high performing models that extract relations from New York Times Corpus (Riedel, Yao, and McCallum 2010) and TACRED challenges (Zhang et al. 2017) respectively. Hyponyms represent hierarchical connection between entities of a domain and represent important relationships. For instance, a well-known work by (Hearst 1992) uses syntactic patterns such as [*Y such as A, B, C*], [*Y including X*], or [*Y, including X*] to extract hyponyms. Our goal is to extract preliminary hyponym relations from the concepts extracted by the CR and to connect the entities through verb phrases.

CONCEPT RECOGNITION

SE concepts are less ambiguous as compared to generic natural language text. A word usually means one concept. For example, the word *'system'* usually means the same when referring to a *'complex system'*, *'system structure'*, or *'management system'* in the SE domain. In generic text, the meaning of terms like *'evaluation'*, *'requirement'*, or *'analysis'* may contextually differ. We would like domain specific phrases such as *'system evaluation'*, *'performance requirement'*, or *'system analysis'* to be single entities. Based on the operational and system concepts described in (Krishnan, Coronado, and Reed 2019), we carefully construct a set of concept-labels for the SE handbook which is shown in the next section.

BIO Labelling Scheme

1. ***abb***: represents abbreviations such as *TRL* representing *Technology Readiness Level*.
2. ***grp***: represents a group of people or an individual such as *Electrical Engineers*, *Systems Engineers* or a *Project Manager*.
3. ***syscon***: represents any system concepts such as *engineering unit*, *product*, *hardware*, *software*, etc. They mostly represent physical concepts.
4. ***opcon***: represents operational concepts such as *decision analysis process*, *technology maturity assessment*, *system requirements review*, etc.
5. ***seterm***: represents generic terms that are frequently used in SE text and those that do not fall under *syscon* or *opcon* such as *project*, *mission*, *key performance parameter*, *audit* etc.
6. ***event***: represents event-like information in SE text such as *Pre-Phase A*, *Phase A*, *Phase B*, etc.
7. ***org***: represents an organization such as *'NASA'*, *'aerospace industry'*, etc.
8. ***art***: represents names of artifacts or instruments such as *'AS1300'*
9. ***cardinal***: represents numerical values such as *'1'*, *'100'*, *'one'* etc.
10. ***loc***: represents location-like entities such as *component facilities* or *centralized facility*.
11. ***mea***: represents measures, features, or behaviors such as *cost*, *risk*, or *feasibility*.

Abbreviations

Abbreviations are used frequently in SE text. We automatically extract abbreviations using simple pattern-matching around parentheses. Given below is a sample regex that matches most abbreviations in the SE handbook.

```
r"\([ ]*[A-Z][A-Za-z]*[ ]*\)"
```

An iterative regex matching procedure using this pattern over the preceding words will produce the full phrase of the abbreviation. *'A process to determine a system's technological maturity based on Technology Readiness Levels (TRLs)'* produces the abbreviation ***TRL*** which stands for ***Technology Readiness Levels***. *'Define one or more initial Concept of Operations (ConOps) scenarios'* produces the abbreviation ***ConOps*** which stands for ***Concept of Operations***. We pre-label these abbreviations as concept entities. Many of these abbreviations are also provided in the Appendix section of the handbook which is also extracted and used as concepts.

Common-Knowledge Definitions

Various locations of the handbook and the glossary provide definitions of several SE concepts. We collect these and compile a comprehensive definitions document which is also used for the concept recognition task. An example definition and its description is shown below:

Definition*: Acceptable Risk*

Description*: The risk that is understood and agreed to by the program/project, governing authority, mission directorate, and other customer(s) such that no further specific mitigating action is required.*

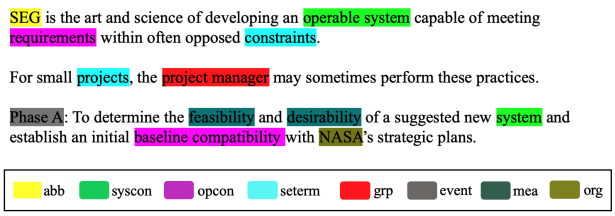


Figure 2: A Snippet of the concept-labelled dataset

O	73944	B-cardinal	414	I-grp	132
B-opcon	5530	B-abb	354	B-org	87
B-syscon	1640	B-event	350	I-seterm	26
B-seterm	1431	I-event	218	B-art	17
I-opcon	1334	I-syscon	201	I-org	12
B-mea	1117	I-abb	156	I-loc	3
B-grp	499	I-mea	145	B-loc	2

Table 1: Unique Tag Count from the CR dataset

CR Dataset Construction and Pre-processing

Using python tools such as *PyPDF2*, *NLTK*, and *RegEx* we build a pipeline to convert PDF to raw text along with extensive pre-processing which includes joining sentences that are split, removing URLs, shortening duplicate non-alpha characters, and replacing full forms of abbreviations with their shortened forms. We assume that the SE text is free of spelling errors. For the CR dataset, we select coherent paragraphs and full sentences by avoiding headers and short blurbs. Using domain keywords and a domain expert, we annotate roughly 3700 sentences at the word-token level. An example is shown in Figure 2 and the unique tag count is shown in Table 1.

Fine tuning with BERT

Any language model can be used for the purpose of customizing an NER problem to CR. We choose to go with BERT (Devlin et al. 2018) because of its general-purpose nature and usage of contextualized word embeddings.

In the hand-labelled dataset, each word gets a label. The idea is to perform multi-class classification using BERT's pre-trained cased language model. We use pytorch transformers and hugging face as per the tutorial by (Huang 2019) which uses $BertForTokenClassification$. The text is embedded as tokens and masks with a maximum token length. This embedded tokens are provided as the input to the pre-trained BERT model for a full fine-tuning. The model gives an F1-score of 0.89 for the concept recognition task. An 80-20 data split is used for training and evaluation. Detailed performance of the CR is shown in Table 2 and 3. Additionally, we also implemented CR using spaCy (Honnibal and Johnson 2015) which also produced similar results.

RELATION EXTRACTION

In this work, for relation extraction, we focus on hyponyms and verb phrase chunking. Hyponyms are more specific concepts such as *earth* to *planet* or *rose* to *flower*. Verb phrase chunking connects the named entities recognized by the CR model through verbs.

	precision	recall	f1-score	support
syscon	0.94	0.89	0.91	320
opcon	0.87	0.91	0.89	1154
seterm	0.98	0.94	0.96	287
mea	0.91	0.90	0.90	248
grp	0.94	0.93	0.94	89
org	1.00	0.11	0.21	26
cardinal	0.90	0.92	0.91	71
event	0.71	0.78	0.76	77
abb	0.82	0.58	0.68	79
art	0.00	0.00	0.00	4
loc	0.00	0.00	0.00	1
micro/macro-avg	**0.90**	**0.88**	**0.88**	**2356**

Table 2: Performance of different labels

F1-Score	Accuracy	Accuracy without 'O'-tag
0.89	0.97	0.86

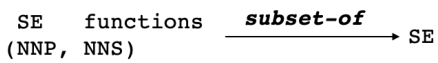
Table 3: Overall Performance of CR; For fairness, we also provide the accuracy when the most common 'O'-tag is excluded from the analysis.

Hyponyms from Definitions

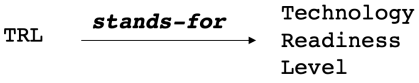
The definition document consists of 241 SE definitions and their descriptions. We iteratively construct entities in increasing order of number of words in the definitions with the help of their parts-of-speech tags. This helps in creating *subset-of* relation between a lower-word entity and a higher-word entity. Each root entity is lemmatized such that entities like *processes* and *process* appear only once.

Hyponyms from POS tags

Using the words (especially nouns) that surround an already identified named entity, more specific entities can be identified. This is performed on a few selected entity tags such as *opcon* and *syscon*. For example, consider the sentence *'SE functions should be performed'*. *'SE'* has tag *NNP* and *'functions'* has tag *NNS*. We create a relation called ***subset-of*** between *'SE functions'* and *'SE'*.



Relations from Abbreviations



Relations from abbreviations are simple direct connections between the abbreviation and its full form described in

the abbreviations dataset. Figure 3 shows a snippet of knowledge graph constructed using *stands-for* and *subset-of* relationships. Larger graphs are shown in the demo.

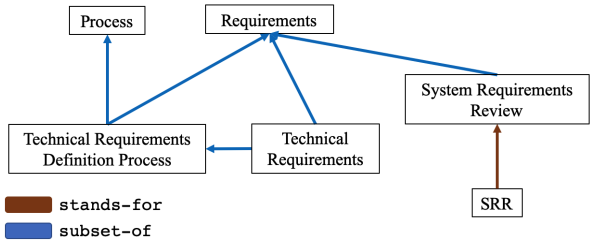


Figure 3: A snippet of the knowledge graph generated

Relation Extraction using Verb Phrase Chunking

Finally, we explore creating contextual triples from sentences using all the entities extracted using the CR model and entities from definitions. Only those phrases that connect two entities are selected for verb phrase extraction. Using NLTK's regex parser and chunker, a grammar such as

```
VP: {(<MD>|<R.*>|<I.*>|<VB.*>|<JJ.*>|
<TO>)*<VB.*>+(<MD>|<R.*>|<I.*>|<VB.*>|
<JJ.*>|<TO>)*}
```

with at least one verb, can extract relation-like phrases from the phrase that links two concepts. An example is shown in Figure 4. Further investigation of relation extraction from SE handbook is left as future work.

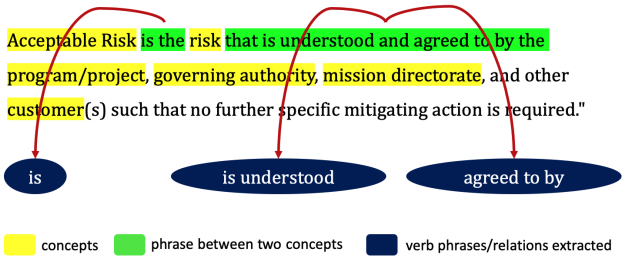


Figure 4: Relation Extraction using Verb Phrase

CONCLUSION AND FUTURE WORK

We presented a common-knowledge concept extractor for the Systems Engineer's Virtual Assistant (SEVA) system and showed how it can be beneficial for downstream tasks such as relation extraction and knowledge graph construction. We construct a word-level annotated dataset with the help of a domain expert by carefully defining a labelling scheme to train a sequence labelling task to recognize SE concepts. Further, we also construct some essential datasets from the SE domain which can be used for future research. Future directions include constructing a comprehensive *common-knowledge* relation extractor from SE handbook and incorporating such human knowledge into a more comprehensive machine-processable *commonsense* knowledge base for the SE domain.

References

AI2 Allen Institute for AI. Aristo: An intelligent system that reads, learns, and reasons about science. https://allenai.org/aristo/. Accessed: 2019-08-12.

Baevski, A.; Edunov, S.; Liu, Y.; Zettlemoyer, L.; and Auli, M. 2019. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elaasar, M. 2019. Open casesar; the case for integrated model centric engineering.

Etzioni, O.; Fader, A.; Christensen, J.; Soderland, S.; and Mausam. 2011. Open information extraction: the second generation. *IJCAI* 3–10.

Hart, L. E. 2015. Introduction to model-based system engineering (mbse) and sysml. http://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf. Accessed: 11-09-2017.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Vol. 2*, 539–545. ACL.

Honnibal, M., and Johnson, M. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1373–1378. ACL.

Huang, B. 2019. Ner with bert in action.

JPL. 2016. Imce ontological modeling framework.

Krishnan, J.; Coronado, P.; and Reed, T. 2019. Seva: A systems engineer's virtual assistant. In *AAAI-MAKE Spring Symposium*.

NASA. 2017. Nasa systems engineering handbook.

Panton, K.; Matuszek, C.; Lenat, D.; Schneider, D.; Witbrock, M.; Siegel, N.; and Shepard, B. 2006. *Common Sense Reasoning – From Cyc to Intelligent Assistant*. Berlin, Heidelberg: Springer Berlin Heidelberg. 1–31.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 148–163. Springer.

Sang, E. F., and De Meulder, F. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Soares, L. B.; FitzGerald, N.; Ling, J.; and Kwiatkowski, T. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Xu, P., and Barbosa, D. 2019. Connecting language and knowledge with heterogeneous representations for neural relation extraction. *arXiv preprint arXiv:1903.10126*.

Zellers, R.; Bisk, Y.; Schwartz, R.; and Choi, Y. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *CoRR* abs/1808.05326.

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*, 35–45.