Rosoka's Hybrid AI Approach

Gregory F. Roberts, Founder & CEO
Michael A. Sorah, Founder & CTO
[groberts,msorah]@rosoka.com

Abstract

This paper describes how Rosoka uses AI to expand the language models during both the model training development phase and during production processing. This approach is an interplay between DKR, CKR, and RKE that provides for a high fidelity NLP system. Dynamic learning during production operation provides the ability to deal with unforeseen and novel entities during production operations. The amount of training necessary during the modeling phase is lowered, thereby reducing TCO.

Introduction

Artificial Intelligence (AI) algorithms are useful tools to expand the recall availability of a rule-based, natural language processing (NLP) entity extraction system. There are primarily two AI injection points for augmenting the entity extraction system: during the building of linguistic models and while the system is running in production. Both of these approaches are non-exclusive and can be used in concert to improve the overall performance of an extraction system. The purpose of this paper is to provide an understanding of the three types of knowledge that an NLP system encodes and describe the approach of how Rosoka NLP engine uses AI to encode these knowledge types.

Knowledge Encoding

An NLP system has to have a way to *remember* and *know* pieces of information for it to make its evaluations. Typically, an NLP system will encode these data in dictionaries, patterns, and templates. There are three types of knowledge that can be encoded: rote knowledge, compositional knowledge, and dynamic knowledge.

Rote Knowledge

Rote Knowledge Encoding (RKE) is the ability to take real world knowledge and add that to an NLP system. This process is typically done by adding entries to dictionaries or lexicons. This type of knowledge encoding sometimes has a bad reputation as *just hard-coded* knowledge. RKE enables the system to encode high value, low occurrence entities that are difficult for other statistical model training to encode. RKE is the most precision-focused information that is integrated in an NLP tool. These data are the real world information nuggets that the knowledge engineer or linguist building the system just *knows* and wants the NLP tool to *know*. For instance, the knowledge engineer's favorite movie is *Raising Arizona* and their favorite actors are Nicolas Cage, Holly Hunter, and John Goodman.

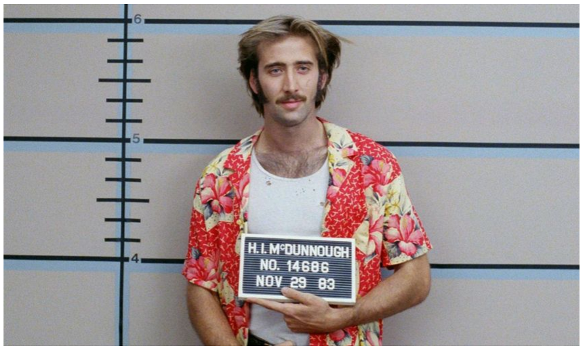

Figure 1. Nicolas Cage playing H.I. McDunnough in the 20th Century Fox movie Raising Arizona (1987).

The knowledge engineer just knows these names, once they have added them to an NLP system's dictionary, the NLP system will just know the names too. They do not have to think about it or decompose the names into their constituent given name and surname pieces. Adding in lists of known information to an NLP tool is RKE.

Compositional Knowledge

Compositional Knowledge Retention (CKR) is the process of using template or canonical patterns along with labeled dictionaries or lexicons to expressly extract names or other key pieces of information for an NLP tool. A canonical pattern for a person named entity might be something like a given name that is followed by a surname (Nicolas Cage). Or it could be a capitalized initial followed by a surname (N. Cage). The key to CKR is the understanding of the components in the labeled dictionaries or lexicons and how they relate to the canonical pattern.

Given Name	Surname
Nicolas	Cage
Holly	Hunter
John	Goodman

Table 1. Raising Arizona Actors' Names

Using the given names and surnames in **Table 1**, there are 9 name combinations that could fit the *given name followed by a surname* canonical pattern:

> Nicolas Cage, Nicolas Hunter, Nicolas Goodman, Holly Cage, Holly Hunter, Holly Goodman, John Cage, John Hunter, John Goodman.

In the example canonical pattern of [given name surname] there are $n \cdot m$ possible name combinations; however, when looking at other canonical name forms that include more extensive groups of name forms (multi-part given names, middle names, and patronymic/matronymic name components, etc.) the possible name combinations skyrocket. Each additional given name or surname added to a dictionary enables the system to recognize an additional number of entities based on the formula in **Figure 2**.

$$\frac{(\sum n_i)!}{\prod n_i!} \cdot \frac{(\sum m_j)!}{\prod m_j!}$$

Where n= number of given names, i is then length of given/middle names, and m is the number of surnames, j is the length of combinitoral surnames that make up the canonical name forms.

Figure 2. Equation for the number of name permutations based on known given names and surname combinations.

Using Rosoka's AI approach to discover additional name components, along with hand-compiled dictionaries, is a force multiplier to extend the CKR reach and coverage exponentially.

Dynamic Knowledge

Dynamic Knowledge Retention (DKR) is the process of using an NLP tool to look at a larger linguistic context instead of just a localized canonical pattern. For instance, in the movie Raising Arizona, Nathan Arizona, Sr. was fond of exclaiming, “Or my name ain’t Nathan Arizona!”. One intuitively knows that Nathan Arizona is probably a person's name. In another context, one might assume that Nathan Arizona was a town in the US state of Arizona. But in the movie context of:

Or my name ain’t ______ _______!

It is naturally understood that what comes next is going to be a person’s name, regardless of what is there.

Or my name ain’t Nathan Arizona!
Or my name ain’t Edwina McDunnough!
Or my name ain’t Gale Snoats!
Or my name ain’t Leonard Smalls!

DKR recognizes these productive linguistic contexts and can use this dynamic knowledge for further analysis.

Knowledge Encoding Interplay

Rosoka leverages the interplay between RKE, CKR, and DKR to expand the extraction expressive footprint.

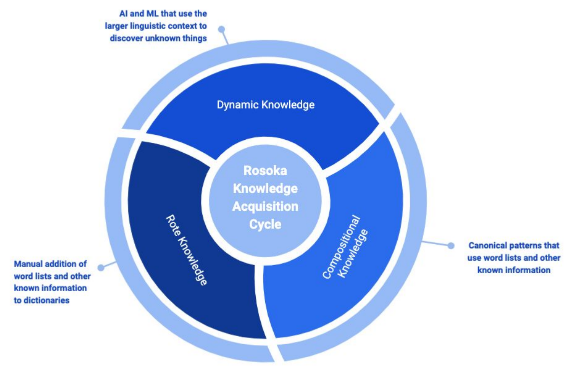

Figure 3. Rosoka uses a combination of RKE, CKR, and DKR for its knowledge acquisition.

Novel entities that are discovered by the DKR process are leveraged as known entities that are incorporated in the RKE. The novel entity can be further deconstructed into the components of the dynamic pattern as part of the CKR process. Each stage of knowledge encoding impacts the other stages to provide additional building blocks.

Learning During Model Training

Linguistic model training for Rosoka is the process whereby a linguist, knowledge engineer, or subject matter expert (modeler) affects the output capabilities of Rosoka, such as modifying entities, entity attributes, and relationships. Once the extraction ontology has been established, the modeler affects the extraction capabilities through dictionaries and rules.

The Rosoka engine follows these high level steps for extraction:

- Language identification is decided for each token in the token sequence and rolled-up for the document.

- A floating window of each token sequence is looked up in the appropriate dictionaries to provide an initial semantic vector mapping.
- The token sequence is evaluated against the set of rules through a floating window of vector sequences to provide an output set of semantic vectors through degree of fitness algorithms.

Model training involves running the engine over a set of data and exploring the output for the modeler to modify Rosoka assets (dictionaries, rules, etc.) and reprocessing the documents and evaluating the differences. Rosoka could use AI through machine learning to provide a force multiplier to the modeling process.

There are two broad categories of machine learning: supervised and unsupervised. Supervised machine learning keeps a person in the loop. Machine learned evaluations are presented to a human supervisor to provide validation of correctness. Validated evaluations can be added to the language model, thereby improving the language model. Unsupervised machine learning uses a confidence metric to decide and validate correctness without the human supervisor. Learned evaluations that exceed a predefined confidence metric are automatically added to the language model.

Historically, AI has been used as the processing engine for NLP systems. [3],[6] Rosoka's approach is to use AI as a confidence measure to decide what elements should be added or removed from the linguistic model.

Rosoka takes a supervised machine learning approach during the modeling phase for dictionary expansion. The training output vector is compared against known, possible input vectors to evaluate for newness and novelty. Is the output vector something already known or is the output vector something that the model already knows about. If the output vector has a vector value which is not a known possibility it is then evaluated for correctness. The supervisor can evaluate the new condition as correct or evaluate the condition as incorrect. These decisions are then translated to the language model. **Figure 4** shows a Rosoka Studio screen capture of the supervised machine learning interface. Within Rosoka Studio this process is called Machine Discovery.

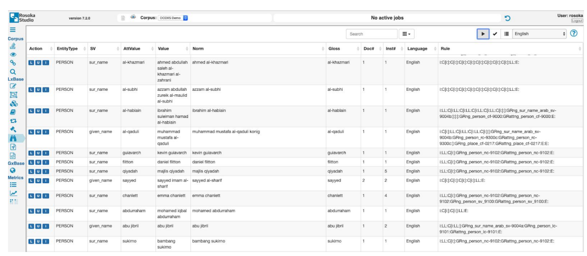

Figure 4. Screenshot of Rosoka Studio Machine Discovery Evaluator Page for Supervised Machine Learning.

The supervisor decides the correctness of the machine learned output by selecting to learn (L), unlearn (U), or ignore (I) what the machine learning algorithms provided.

Learning During Production Processing

Up to this point, we have described how Rosoka uses AI through supervised machine learning during the modeling phase of development. Rosoka can also use unsupervised AI during operations, after the modeling process. As previously discussed, Rosoka's approach uses AI to decide what elements should be added or removed from the linguistic model.

Rosoka AI uses a modified General Regression Neural Net (GRNN) to learn novel semantic vectors discovered by the algorithms from ideal rule traces to automatically update the language model on the fly, during operations with dynamic knowledge.

A GRNN is a special case of the Radial Basis Function Network (RBFN) that is itself a form of kernel regression networks.[5] A GRNN does not require an iterative training procedure as is the case for back propagation networks. Al-Mahasneh, et al. state that feedforward, back-propagation networks suffer from sensitivity to randomly assigned initial weights.[1] A GRNN does not suffer from this problem because it uses the target value as a weight connection between the pattern and the summation layer neurons instead of using randomized weights. A GRNN provides an estimate of continuous variables by converging into either a linear or nonlinear regression surface. This type of artificial neural network (ANN) has a simple yet powerful structure of four layers: an input layer, a pattern layer, a summation layer, and an output layer. GRNN estimates the output by using a weighted average of the outputs compiled from the training dataset. An output weight is computed by using a distance measure between the training data and test data. If the distance between the training data and test data is small, then additional weight is allocated to the output, otherwise less weight is allocated to the output.

The activation function of the GRNN is as follows:

$$y_i = \frac{\sum_{i=1}^{n} g_i . w_{ij}}{\sum_{i=1}^{n} g_i}$$

where w_{ij} is the weighting functions on the input training vectors. The output of the hidden layer neurons is g_i.

Figure 5. Activation function for GRNN.

The GRNN can be used for NLP tagging directly even for sparsely populated training data such as *low resource* languages.[6]

Figure 6 shows how the GRNN network is coupled to the output of the rule based extraction engine. When novel information is discovered, those data are compared against the neural network node states to establish the confidence measure; thereby providing the degree of fitness. A comparator is used to establish if these data are above or below the threshold for inclusion in the language model as dynamic feedback during operations.

A system that uses the Rosoka engine with the Rosoka AI continually updates the language model as each document is processed and new information is learned. Subsequent documents processed leverage the updated language model as part of the operational processing stream.

By using the comparator of the output of the rules engine to that of the neural network mitigation of learning *drift* is achieved allowing the learning to be applied dynamically during production usage of the learning feedback system. In essence, the comparator couples the high precision of the rule based extraction, with the expanding recall provided by the unsupervised AI. Persistence of node states allows production shutdown, startup and restart without retraining the system.

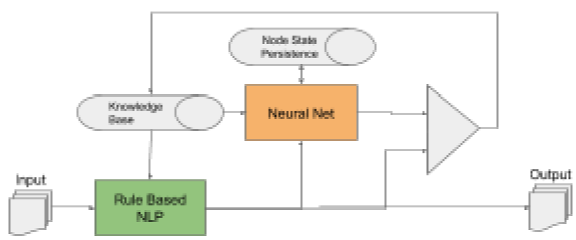

Figure 6. Feedback loop coupling provides for DKR during operational production use.

The feedback loop enables Rosoka to incorporate the DKR learned from AI into the language model with the existing RKE and CKR in real time.

Traditional NLP systems take a lot of linguistic resources to run effectively. A pure AI NLP system requires a statistically significant amount of training data along with hand-tagged examples to learn from. A pure expert rule based system requires a significant amount of training data along with the time it takes to create dictionaries. Therefore, modeling low resource languages with traditional NLP systems become cost-prohibitive because there are no training data available. Rosoka's hybrid AI approach learns as data are encountered. The modeling phase needs much less linguistic resources. The initial language model essentially is the seed for AI expansion.

The ability to detect and learn high value, low probability information without *a priori* training data is significantly increased through Rosoka's AI approach. This allows you to better react to *black swan* events in real time without having to remodel the system in an out-of-band or off-line sequence.

Unsupervised learning in a production environment decreases the amount of model training necessary during the initial modeling phase. Learning is done during production running. This significantly reduces schedule and cost to provide a more cost-effective Total Cost of Ownership (TCO) for continued operations.

Conclusion

This paper described how Rosoka uses AI to expand the language models during both the model training development phase and during production processing.

DKR learned from AI is incorporated into the language model through a feedback loop that merges these data with the existing CKR and RKE in real time.

Rosoka uses supervised learning during the language modeling phase and unsupervised learning during production processing phase to decide what elements should be added or removed from the linguistic model.

Rosoka's hybrid expert trained rules engine and AI approach provides several significant advantages over pure NLP AI or pure expert trained rule engine approaches. The amount of statistically significant training data is reduced. The dynamic learning in operations provides the ability to deal with unforeseen and novel entities during production operations. The amount of training necessary during the modeling phase is lowered, thereby reducing TCO.

References

[1] Al-Mahasneh, A., Anavatti, S., and Garratt, M. (2018). Review of Generalized Regression Neural Networks in Identification and Control of Dynamic Systems. *arXiv:1805.11236 [cs.NE]*.

[2] Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, vol. 2, no. 5, pp. 359–366.

[3] Lopez, M. and Kalita, J. (2017). Deep Learning applied to NLP. *arXiv:1703.03091v1 [cs.CL]*.

[4] Lu, Siyang (2019). Detecting Anomalies From Big Data System Logs. *Electronic Theses and Dissertations. 6525*. https://stars.library.ucf.edu/etd/6525.

[5] Specht, D. (1991). A general regression neural network. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*. 2. 568-76. 10.1109/72.97934.

[6] Yajnik, A. (2018). ANN Based POS Tagging For Nepali Text. *International Journal on Natural Language Computing*, *7*(3), 13–18.