kare
[knowledgeware]

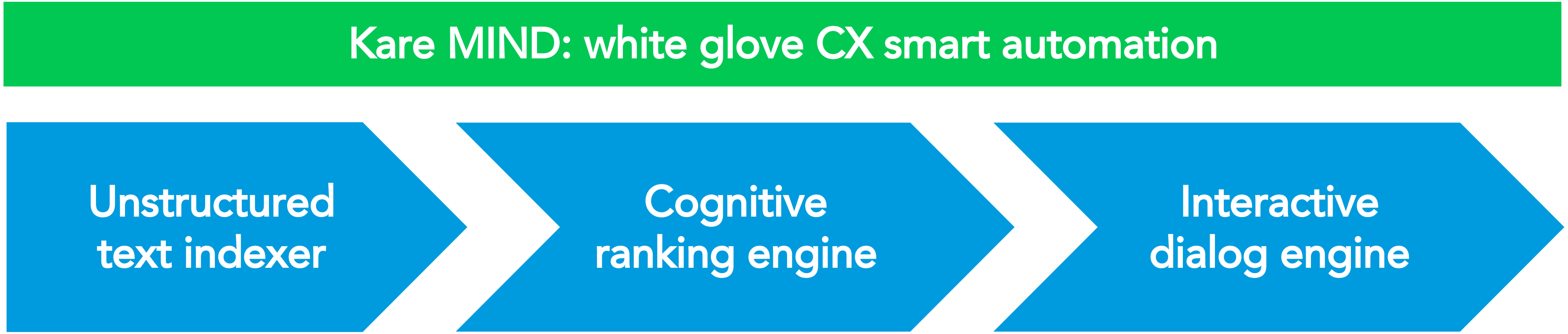
Towards breaking the closed-world assumption in deep neural networks.

Michele Sama / CTO

michele@karehq.com

Jan 28th 2018

Our goal: Discover, answer and learn CX use cases



Kare MIND: white glove CX smart automation

Unstructured text indexer

Cognitive ranking engine

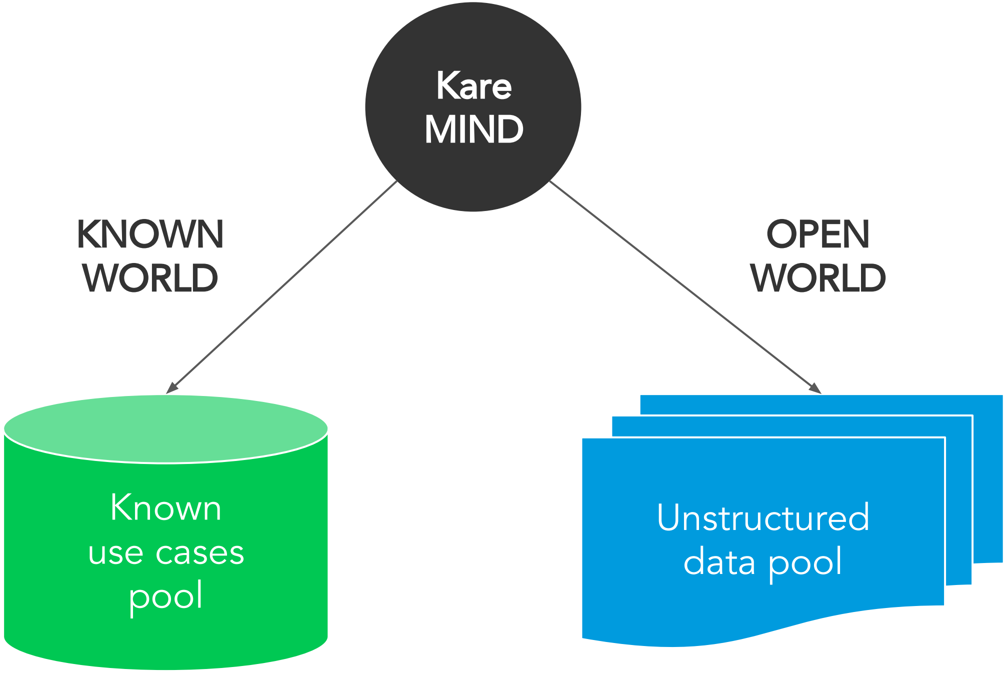
Interactive dialog engine

An **unstructured text indexer** prepares large amounts of unstructured text for contextual and semantic QA

A **cognitive ranking engine** searches through an index including known answers as well as unstructured text

An **interactive dialog engine** answers questions while learning new CX use cases (decision trees)

The desired behaviour: learn from an open world



The desired behaviour is:

1. Use known scenarios when they suffice
   a. Understand when none of the known use cases can be applied

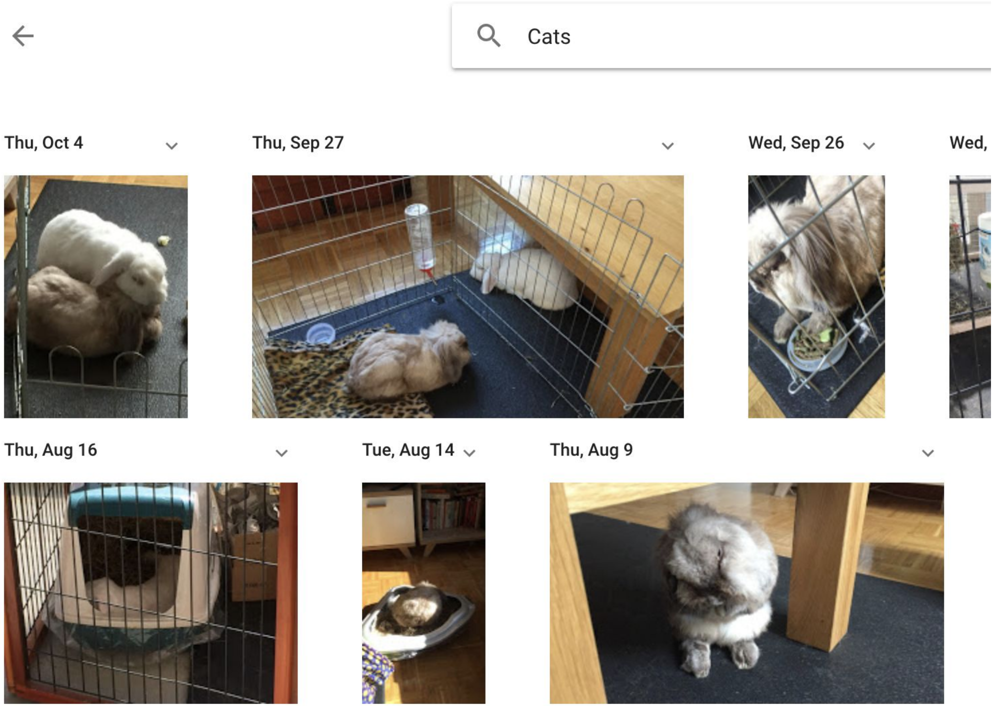
2. Discover new scenarios when necessary
   a. Learn new use cases
   b. Understand when nothing relevant can be discovered

3. Admit defeat when nothing is found

The tech barrier:
Deep NN operates with a closed-world assumption

Classifiers can't say "I don't know"

NN expects to operate with input having the same "features" as the training set



Cats

"What you see is all there is." (D. Kahnemann)

Cognitive bias in humans - we tend to make judgements based on information that is easily available to us.

Breaking the closed-world assumption

Has been a primary research focus.

Has actively driven our choice of technology!

- No free lunch theorem (Wolpert 1996): there is no one ML approach to rule them all, everything is domain-specific. You have to find what works for your specific problem.
- Only a few dozen refs for out of domain in NLP - a very current research problem.

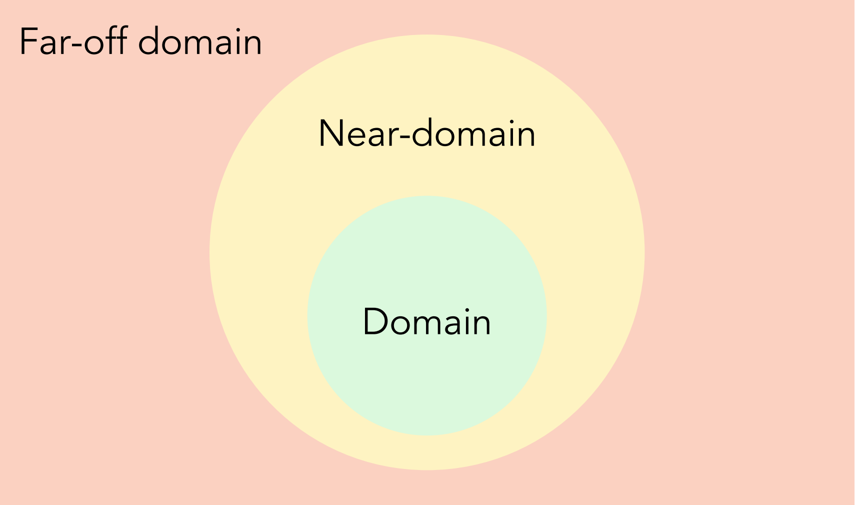
2016: Intent technology (Classification)

- Approaching text comprehension as an open classification problem
- Struggle in domains with 100+ classes.

2018: Cognitive search (Cognitive search)

- Approaching text comprehension as a dense vector graph exploration problem.
- Achieved high prediction quality in domains with 1000-10000 "use cases"

Defining the "Domain"



- **Domain** represent the set of concepts known at training time
- **Near-domain** represents concepts which the model was not trained for but which have a similar set of characteristic as the training data
- **Far-off domain** is the rest of the possible input queries in the wild

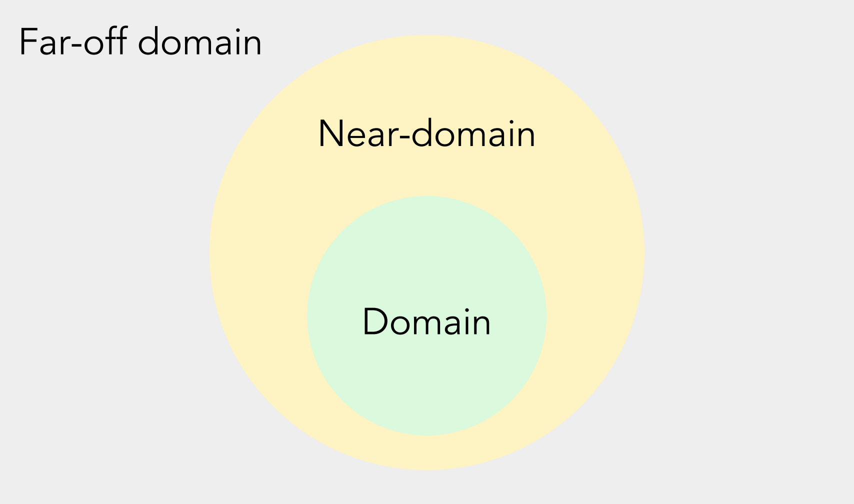
How can new classes be discovered?

Text classification: Catch-all class

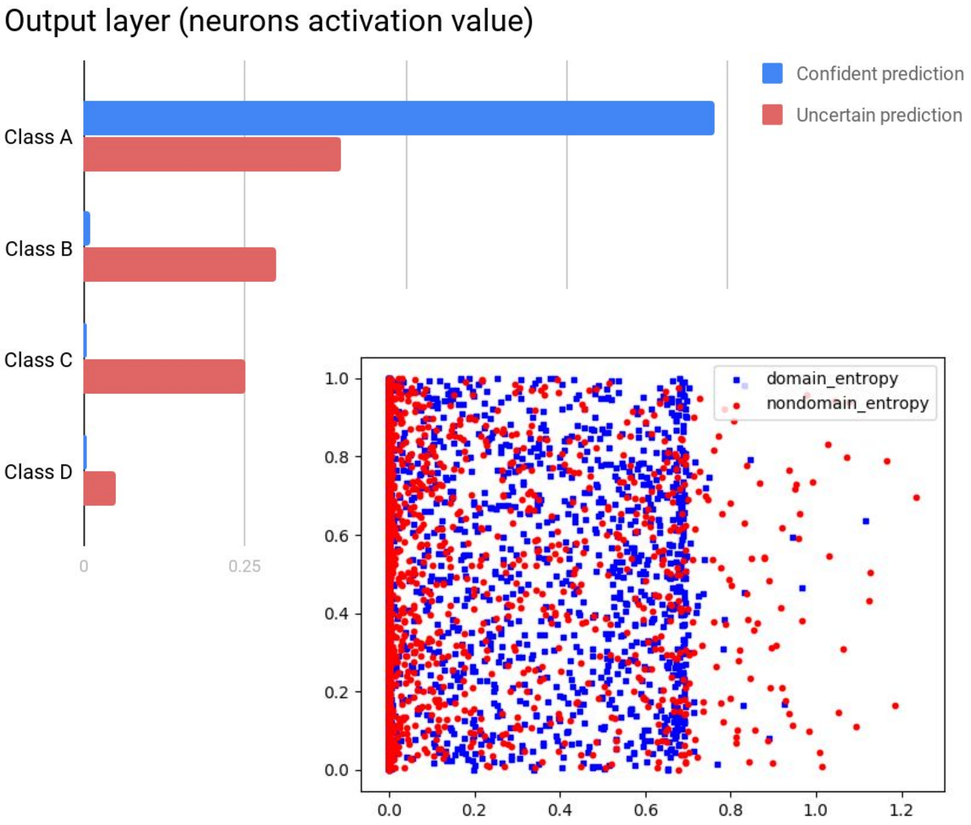
RQ: Can we use a classifier to identify near domain queries?
Yes: with a catch all class.
This approach assumes we can label the 'near domain' examples reliably - how does one know that an out of domain input is 'near domain' in a live system?

On a customer data with 60 classes, 10 were relabeled for the catch all class. We achieved:
Overall: Precision: 0.87, F1: 0.85
Catch-all class: Precision: 0.82, F1: 0.86

Tried both word embeddings and character trigrams as input. Best results with word embeddings.



Text classification: leveraging information theory



When predicting far outside of the training distribution, the classifier gives mixed-class results.

That happens when:

1. The classifier is uncertain (poor training)
2. The query has more than one correct answer (ambiguous query)
3. *The query is a new class*

RQ: Is it possible to exploit the output layer to detect queries out of domain?

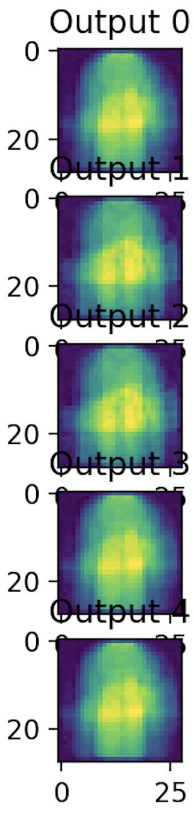
We can understand when the classifier is not confident

Cannot distinguish why

Text classification: domain from the training set

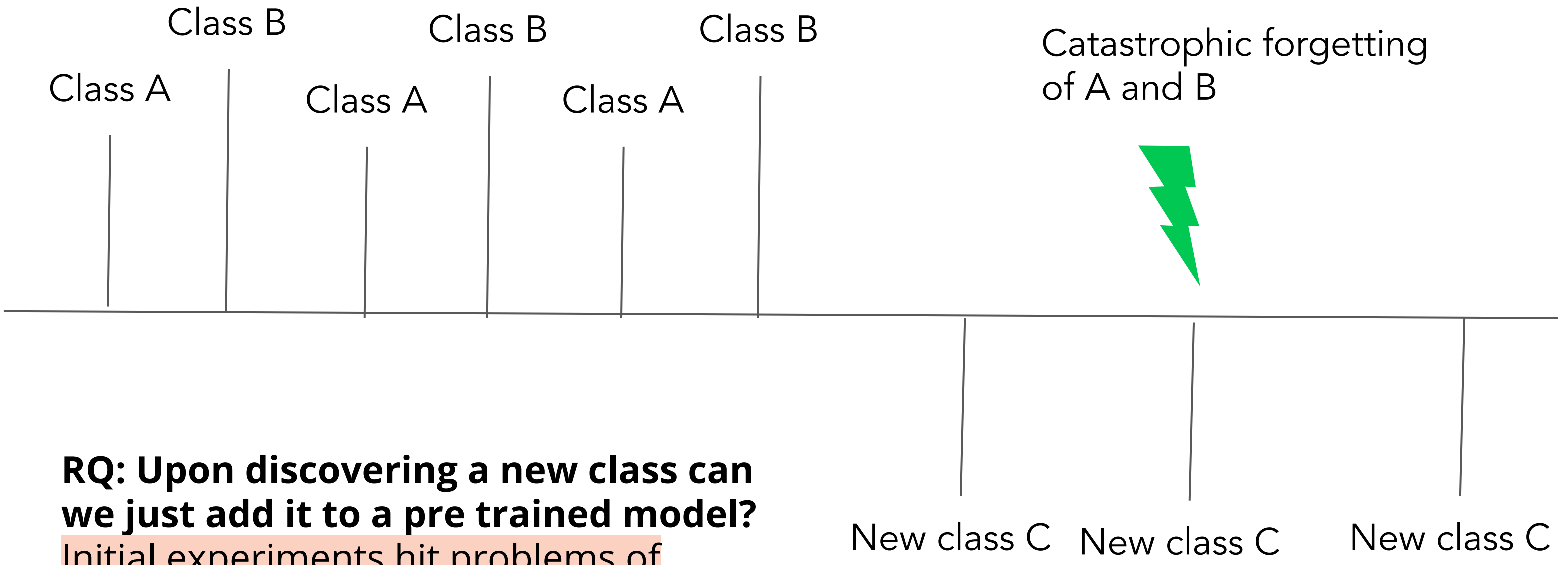
Can we understand if the classifier is operating outside its “area of competence” (i.e. not on what it was trained)?

RQ: Can we model the near domain of a text classifier without counter examples?
Does better than random, worse than the other approaches.
Counter examples make a big difference!



How can a new class be predicted?

Text classification: catastrophic forgetting



Class A

Class B

Class A

Class B

Class A

Class B

Catastrophic forgetting of A and B

New class C

New class C

New class C

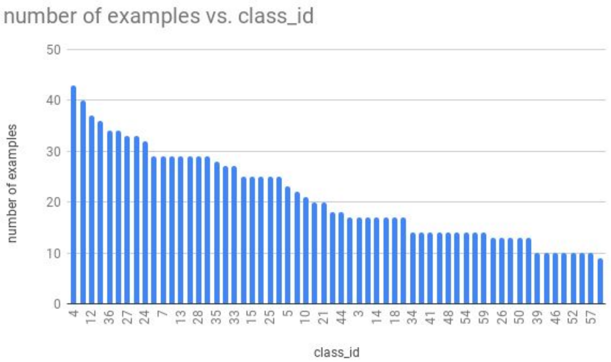
RQ: Upon discovering a new class can we just add it to a pre trained model?

Initial experiments hit problems of catastrophic forgetting

Transfer learning techniques could help.

Text classification: class balance

Class	# samples (customer queries)
Class A Frequent case	50000 :)
Class B Infrequent case	250 :(
Class C Just discovered	5 **o_0**



RQ: can new discovered classes be used for training?
Yes, because it's text and we can use *pre-trained embeddings*

New discovered classes have almost no training data. Often even impossible to do a train/test split.

It is possible to leverage word embeddings and chargrams minimise the amount of required data.

Managed to achieve micro models capable of classifying 50+ classes with:

- at least 5 examples
- upper limit to maintain class balance
- pre-trained embeddings.

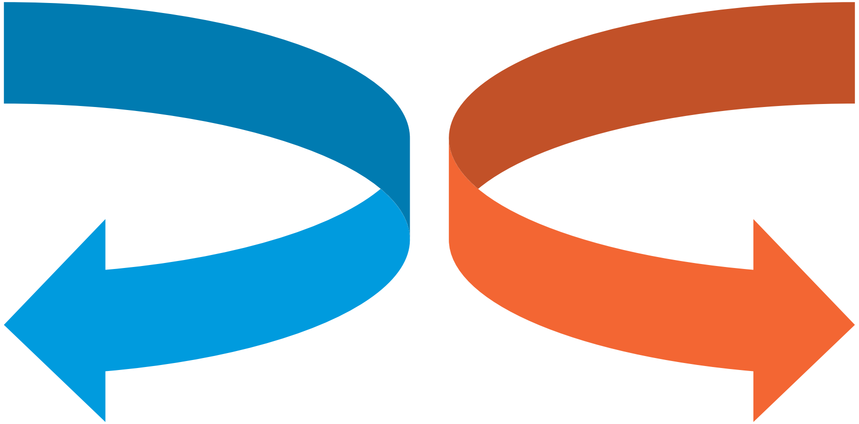
Training time reduced to under 15'

Our solution (without a classifier)

A different approach: Domain mapping

Text understanding. Trained ranking engine numerically quantifies score between query and a possible known node.

Supervised problem



Domain mapping. Knowledge graph being built during use. Graph used to find solution in new domain using given score.

Unsupervised problem

➡ Text understanding allows us to respond to queries from existing knowledge base.

➡ Domain mapping allows us to extend that knowledge base using received queries to generate a new training set, and to label it.

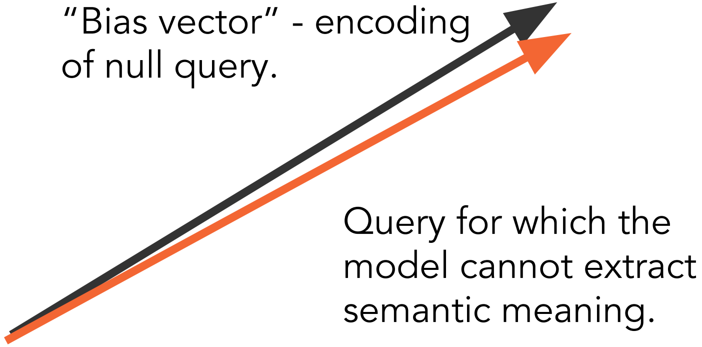
Domain mapping:
identifying text that the model cannot comprehend.

When the model cannot understand the text of a query any derived prediction will be incorrect.

That can happen when:

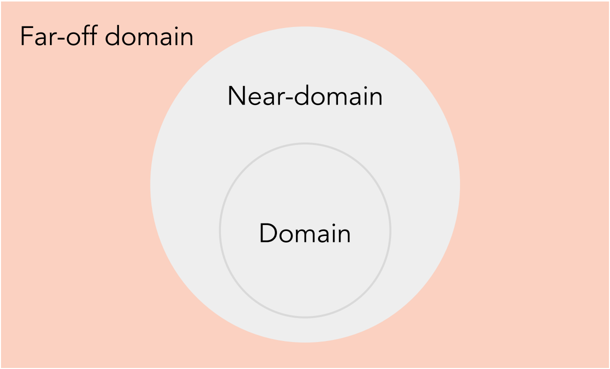
1. The query is in a different language.
2. The query is not a human readable text.
3. The query was too far outside the training domain.

RQ: Can we expand the training domain by looking at runtime data?
Yes by post-processing queries that when encoded carry no information.



"Bias vector" - encoding of null query.

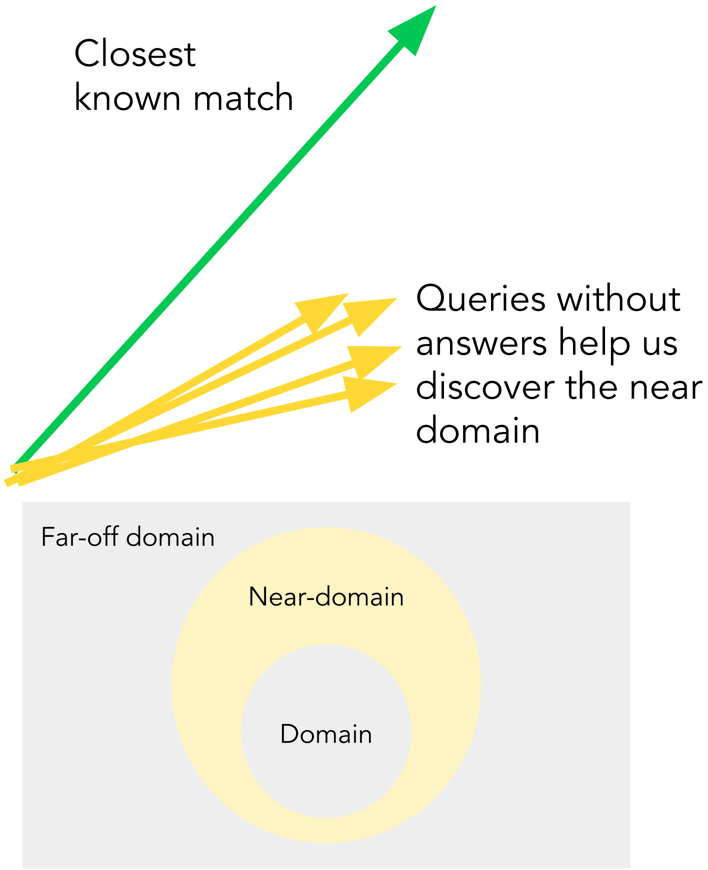
Query for which the model cannot extract semantic meaning.



Graph exploration: identifying new cases in the near domain

The model is capable of encoding the query but there is no existing response in the knowledge graph. The graph needs to be expanded.

RQ: can we dynamically discover new "use cases" and use them without retraining the predictive model?
Yes by expanding the knowledge graph when we identify a "knowledge gap"



Conclusions

1. Classification models are not suitable for problems which do not assume a closed-world
   a. only generalize with the domain of training data
      i. both near-domain and far-off domain remain unpredictable

2. In our solution we divided our NLP task in:
   a. Language inference as an open world ML task
   b. Information retrieval as graph exploration

Try it out: http://karehq.com