Learning neural Question Answering Systems for Low resource Langauges

What’s Question Answering

(@entity3) the @entity2 military claims to have shot down a @entity5 drone , state media reported tuesday . " @entity2 air defense systems shot down a @entity5 @entity9 (unmanned aerial vehicle) north of @entity11 , " the state - run @entity6 said . the @entity5 military lost contact with a mq - 1 predator drone over @entity2 , a @entity5 official said tuesday …

@placeholder 's military claims it shot down a @entity5 drone north of @entity11

(@entity18) – Un hombre fue detenido por su presunta participación en un tiroteo en el que murieron tres personas durante una reunión, en una pequeña ciudad del estado de @entity8, en la costa oeste de @entity5, dijeron las autoridades. En el tiroteo -registrado en @entity11, a unos 45 kilómetros al norte de @entity13- también resultó herida una persona. @entity0, …

El tiroteo se registró en el poblado de @placeholder

The Research Question

- Large amount of data is needed to train a neural QA system
  - 380K Q&A pairs in the CNN news dataset (machine-generated)
  - 120K Q&A pairs in the Maluuba NewsQA dataset (human-annotated)

- But the data doesn’t exist for many languages

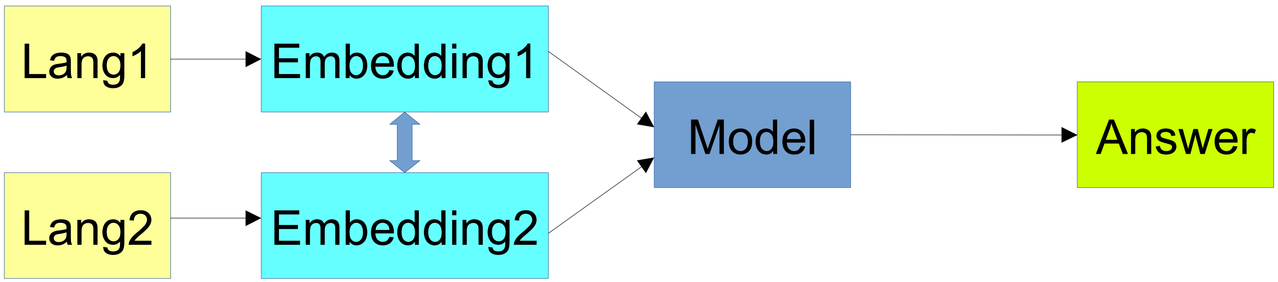
- How to solve this problem?

The IDEA

- Transfer Learning
  - Transfer the learned knowledge from a resource rich language (like English) to a resource poor language (like Spanish)
- Jointly learn the QA models for Engilsh & Spanish
  - Learn abstract knowledge which can be transferred across languages

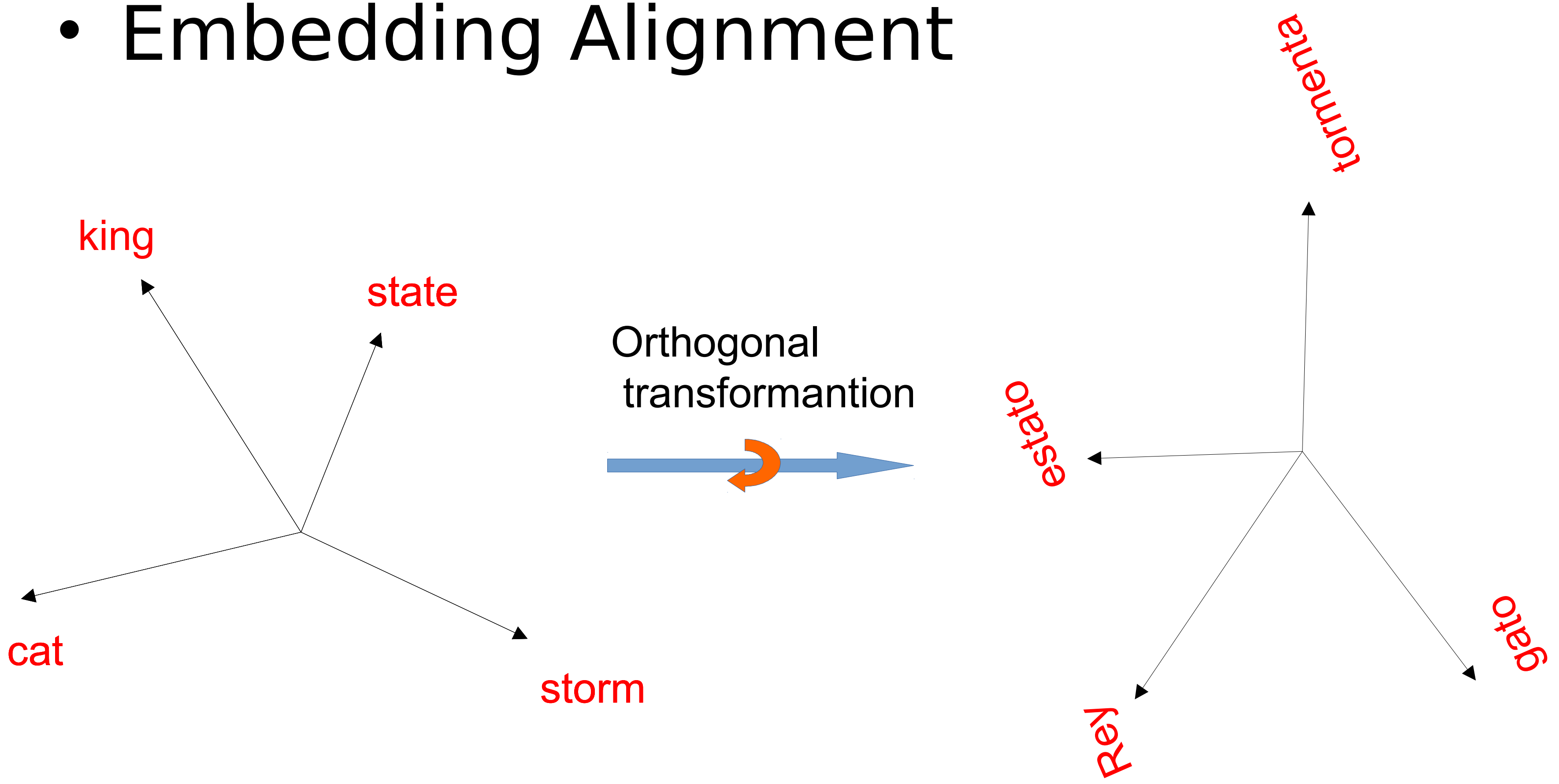
Our Architecture

- Approach 1 – Direct Transfer
  - Train a model in Language 1, then fine-tune the model on a smaller dataset of Language 2
  - Assuming that some aspect of the model can be shared
  - To boost performance, we use aligned word embeddings



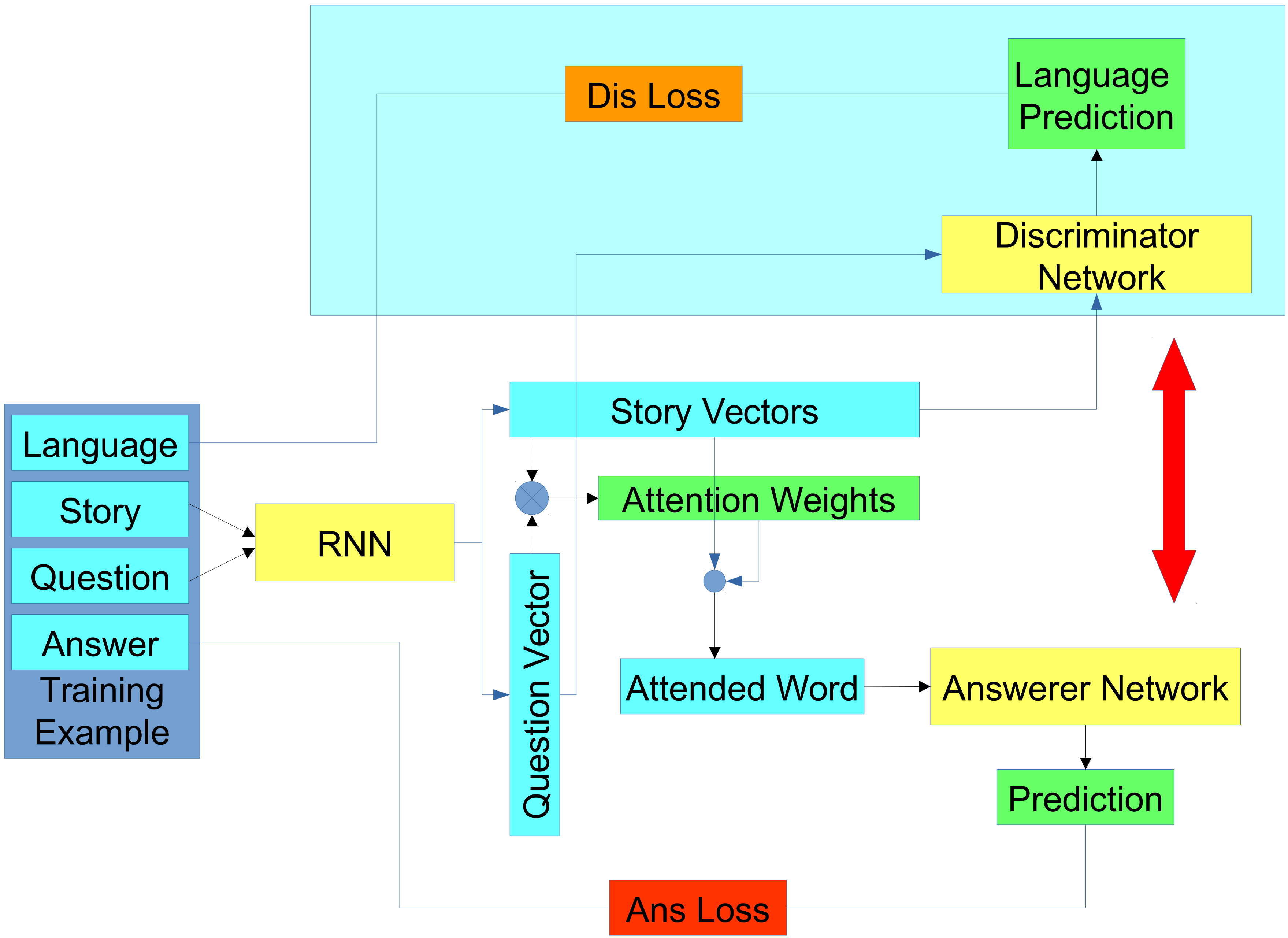
Our Architecture

- Embedding Alignment



Our Architecture

- Approach 2 – Joint Training
  - Simultaneously train the model on both languages
  - Challenge: different optimal weights for each language. Model might learn a bad “average” model
  - Proposed solution: use adversarial training to force the model to learn language-independent features



Dis Loss
Language
Prediction
Discriminator
Network
Language
Story
Question
Answer
Training
Example
RNN
Story Vectors
Attention Weights
Question Vector
Attended Word
Answerer Network
Prediction
Ans Loss

Training the Architecture

- Embedding Alignment Objective:

$$\underset{P}{argmin} \sum_i \| X_i P - Y_i \|$$ For all aligned words

- Adversarial Training Objective

$$\underset{\Theta_A, -\Theta_D}{argmin} [-logP(A|\Theta_A) + logP(L|\Theta_A, \Theta_D)]$$

Follow the negative gradient of theta_D so that we minimise discrim. network loss wrt to it

Mimimise answerer network loss

Maximise discriminator Network loss

Experiments

- Datasets used:
  - English QA data
  - https://github.com/deepmind/rc-data/ (Hermann et al., NIPS 2015)
  - 380298 training QA pairs, 3924 dev and test QA pairs
  -
  - Spanish QA data
  - Collected from www.elmondo.es and CNN Spanish (via cached links on Wayback Machine)
  - Processed by Google Natural Language API (named entity recognition)
  - 69289 training QA pairs, 3839 dev and test QA pairs

Experiments

- Embeddings used:
  - FastText by Facebook
  - https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
    (P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information)
  - 300 dimension word vectors trained on Wikipedia text.
- Word Alignment Dictionary
  - http://opus.lingfil.uu.se/OpenSubtitles2012.php (Jörg Tiedemann, 2012, Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012))
  - Open Subtitles dataset

Results

	En.	Es.	En. → Es. (no training)	En. → Es. (training)
Individual Embeddings				
Mapped Embeddings				
Adversarial Training				
Mapped Embeddings w/ Adversarial Training				

Analysis

Conclusion

Future Work