Placeholder

Placeholder

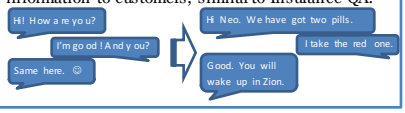
Applying Seq2seq in Building Task-oriented Conversation Systems

Intern: Yiran Luo
Mentor: Lijun Mei, Yipeng Yu
Manager: Shaochun Li

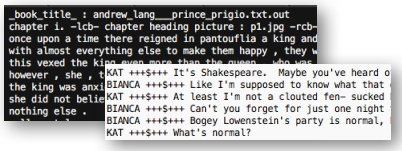
THE OHIO STATE UNIVERSITY

Background

- Logic-based conversation systems in current business solutions can be increasingly complex in structure.
- We aim to build a **simplistic** machine learning model that is able to **learn** from the contents in dialogs.
- We choose **Seq2seq,** which comes with the edge to align up context information in a sequential manner, as the core component.
- Many current applications focus more on building a "chat-bot" to cover generalized topics and mimic human dialogs.
- We choose to build a **task-oriented** conversation system to provide intelligent business specified information to customers, similar to Insurance QA.

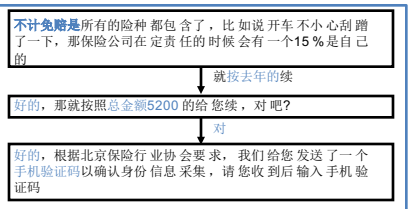


Challenges



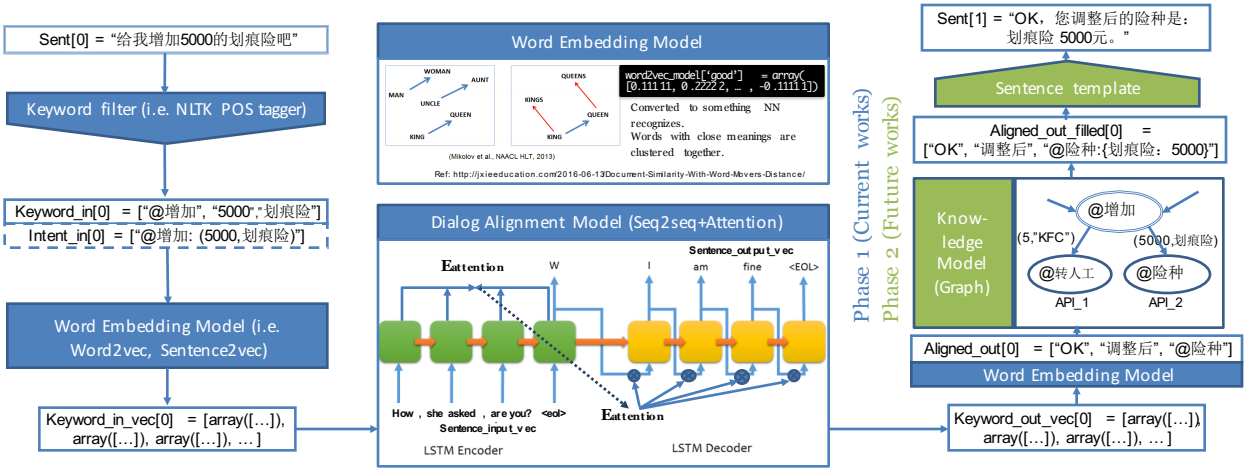
Source: Facebook Children's Book Test (back) and Cornell's Movie-Dialogs Corpus (front)

- The essential data for training within such a designated circumstance are highly restricted. Most of existing training conversations are for open domain purposes.



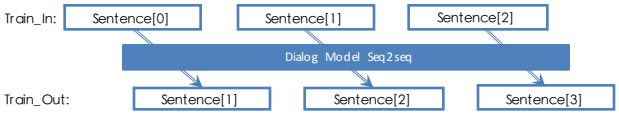
- Open domain data comes with lots of noise. Words such as prepositions are unnecessary in maintaining key information in the context of dialogues.
- Thus we may indefinitely approach our targets, but may never retrieve the exactly same words.

Framework



Experiment & Result

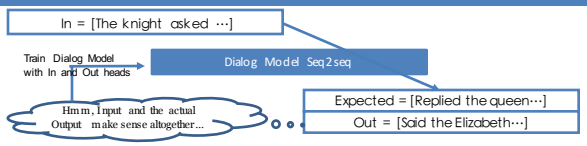
- **Assumptions:** Since few task-oriented dialog data is available at hand, we simulate such a conversation scenario with open domain data. Test sentences are selected from a single story (as a task frame) and we align up every two neighboring sentences while training the Seq2seq model.



- **Experiment settings**: Facebook Children's Book Test Corpus 200k sentences, 200c sentences as test data, 4-layer NN, word vector dimension 100, hidden layer dimension 200, single GPU, 40+ epochs.
  - Word embedding tested with both Word2vec and Sentence2vec. All words are converted to lowercase. Punctuation are limited to " , . .
  - In a single training batch, short sentences are padded with '.' at the end.
  - Implemented in Python with Keras, Theano, gensim and CUDA.
- **Result:** Achieved high similarity while aligning short sentences, but not so well at aligning long sentences so far.
- **Discussion:** In this experiment, we aim at overfitting instead of avoiding it. This may bring down test accuracy given a similar sentence but with slight differences, or given an exactly existing sentence but from another user.

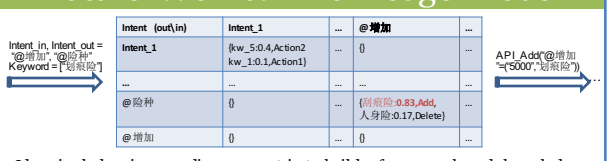
output	target	cos-similarity
``	``	1
forgit	fiddle-de-dee	0.889915
.	!	0.673258
said	replied	0.761018
the	the	1
prince	queen	0.787364
,	;	0.670446
caddy	for	0.58336
honeycomb	the	0.528529
honeycomb	king	0.419499

Innovations



- FasterConvergence: An idea somewhat inspired by reinforcement learning. If we find a pair of good-looking aligned sequences (partial or not), we train the pre-trained Seq2seq model with this very pair for multiple epochs, letting the model recognize the good alignment faster.
- For the Word2vec test, quantitatively, we set a good pair to have an average cosine similarity of over 0.8, with over 3 words in a row. Or when the test case is small (<2000 sentences), we can pick the good pairs out by hand.

Future Works: Knowledge Model



Intent (out\in)	Intent_1	…	@增加	…
Intent_1	{kw_5:0.4,Action2 kw_1:0.1,Action1}	…	{}	…
…	…	…	…	…
@险种	{}	…	{划痕险:0.83,Add, 人身险:0.17,Delete}	…
@增加	{}	…	{}	…

- Ideas include using an adjacency matrix to build a frequency-based knowledge model, and introducing intent keywords to carry user-specific information throughout knowledge retrieval/update.

Placeholder