NLP systems in real-world chatbots: ML operations, practical considerations, language models, explainability

28.04.2022



About me

Larisa Kolesnichenko:
Machine Learning Engineer in Kindly AS
(https://www.kindly.ai).
Background:
Rystad Energy (2019-2021)
Jetbrains research: recommender systems (2017-2019)
Backend developer (Aeroflot airline) (2016-2017)



What Kindly does:

- Chatbots
- Abandonment prediction
- Email solution
- Recommenders

And much more!

Chatbots. What problem do we solve?



We have user requests, we have predefined dialogue sample-reply pairs. What can we do with them?

What we say the problem is: classification!

(**spoiler**: this does not mean this is the only correct way to put the problem!)

What do we want?

Correct answers, less fallbacks (a tradeoff there) and speed
Minimising wrong answers is our focus there

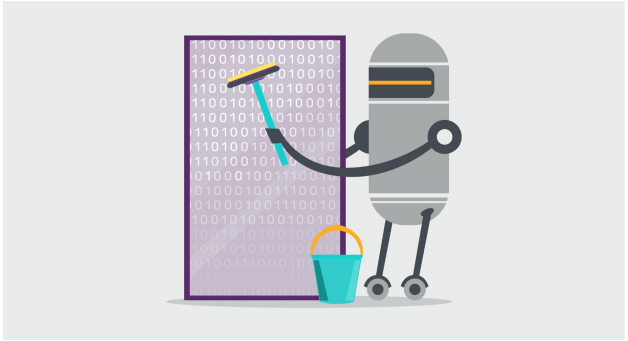




How do we solve the problem



1. Clean the data
2. Pre-process and embed the data
3. Do predictions!



And then… evaluate all things done

Embeddings

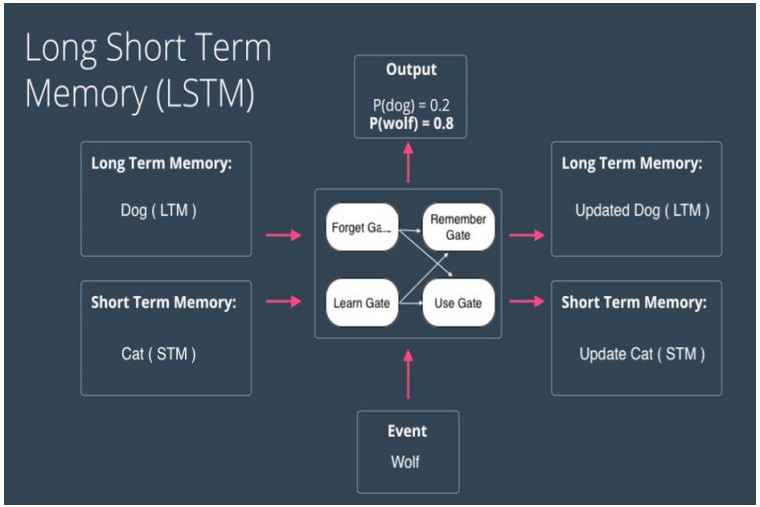
1. Word-level averaged
2. Sentence-level (BERT models. Do not forget there will be some inference time!)

What do we have to predict user intent:

1. Quick and efficient baseline with easy plug-in options

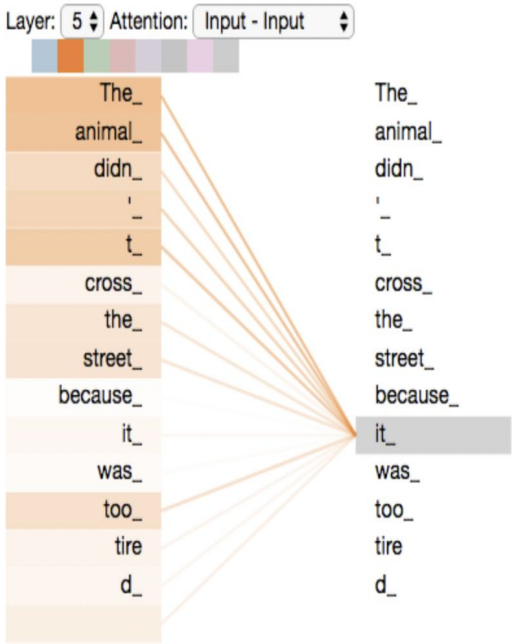
2. ML models zoo:

   Bi-LSTM, CNN with self-attention, MLP and some more





Self-attention example



As we are encoding the word "it" in encoder #5 (the top encoder in the stack), part of the attention mechanism was focusing on "The Animal", and baked a part of its representation into the encoding of "it".

The models are feeding on samples and thrive on them. More good samples -> better training process and better results.

This is why we want to try synthetic text samples generation as well!

Industrial vs academic ML

What do you think the differences might be?

MLOPs

MLOPs - the art of wrapping up your systems with proper alert notifications, abstraction levels, plug-and-play modules

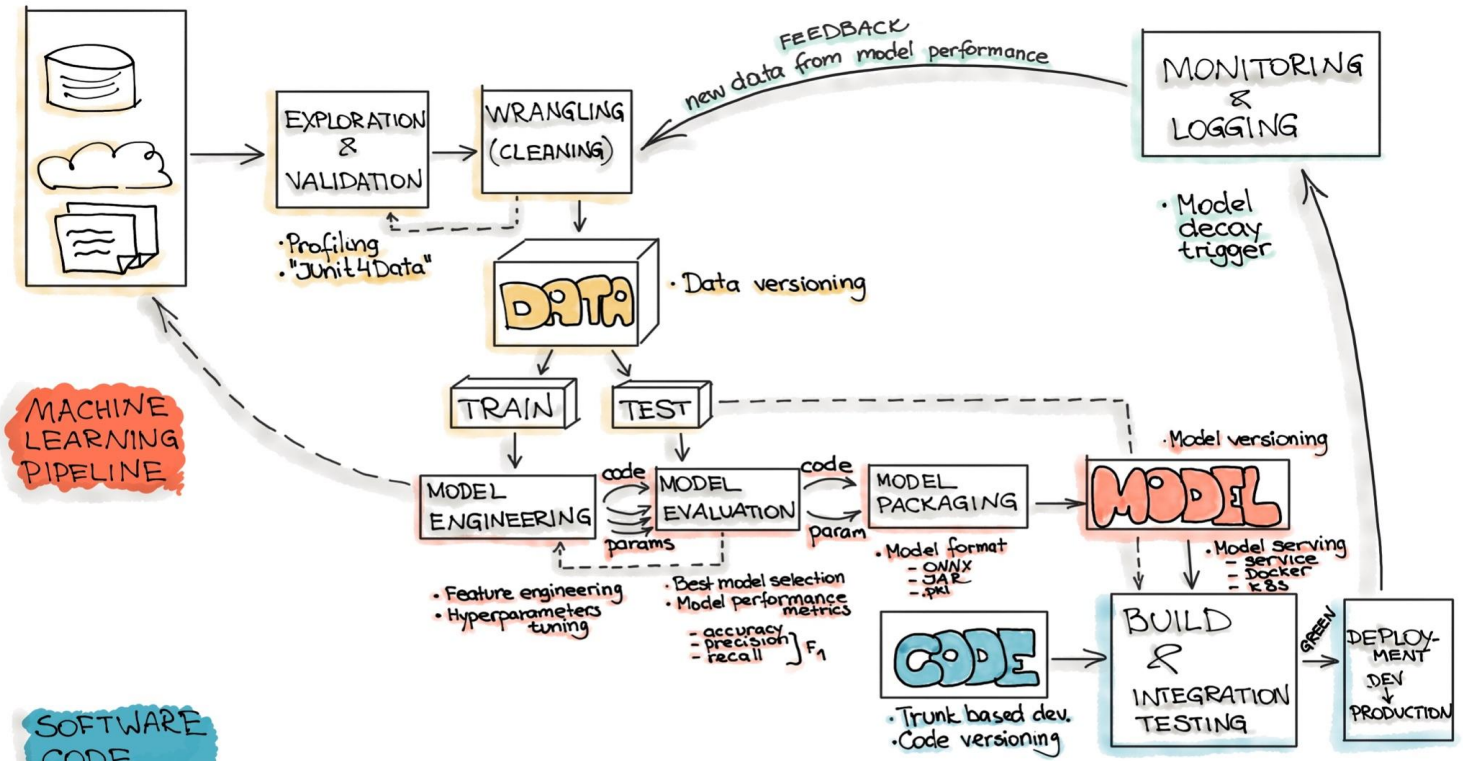
https://ml-ops.org

MLOPs levels:

Level 0 : disconnection between MLOps and Operations
When deployed, we have to actively live monitor the model in production.
Model does not adapt to the latest trends. Big need to experiment often
Level 1 : rapid experimentation, orchestrated experiments, canary deployment configuration, A/B testing Setup
Level 2 : awesomeness. All works without you :D

Progressive delivery concept :
canary releases, A/B testing, bandits, blue/green deployment traffic retouring, smoke testing, multi-armed bandit (dynamically routes the requests to the winning models), contextual bandit (uses ML to learn from the tests results during test)

MACHINE LEARNING ENGINEERING



Unit Tests in the ML WORLD:

- unit tests on data
- unit tests on model performance
- unit tests on None values
- tests on the feature engineering logic : making sure the features are scaled and normalized correctly. Asserting embedding shapes and padding
- tests on the model performance metrics
- tests to avoid inherent bias

How it is in Kindly now:
Something covered by tests, data part needs better coverage.

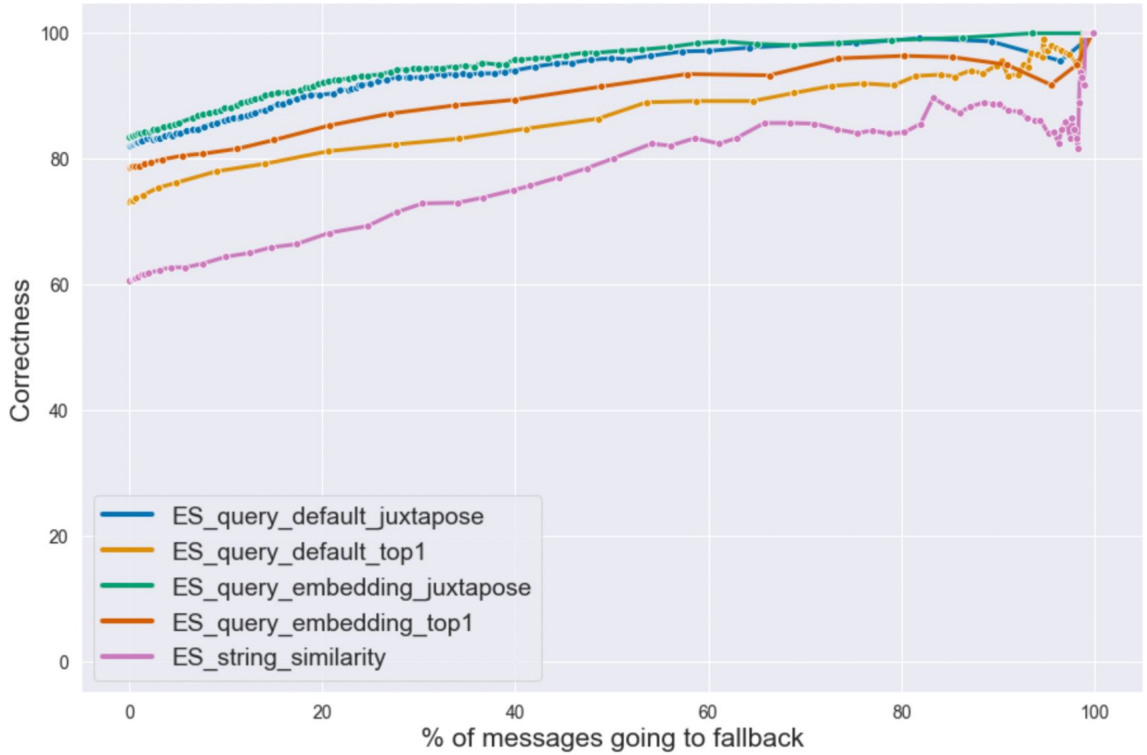
What we can improve here:
Better test coverage, plus we need/want anomaly detection.

Things to watch out for when the model is deployed:

immediate data skews
negative feedback loops
predictive performance
changes in setting data
metrics used during training
data drift (a note: user data usually does not drift fast, enterprise data does)
concept drift
characteristics of features

from the non-functional monitoring, always nice to watch out for
- entanglement: issue when changing anything means changing everything
- observability: how well the internals of a system can be inferred by knowing the inputs and the outputs

When are accuracy numbers not enough. Uncertainty estimation. The easiness to onboard a new bot estimation. Scalability



Practical questions

When to retrain your models? Every night? Every hour? On data update? How much will it cost? When to run the performance evaluation framework?

Online marketplaces usually have heavy overnight models and very very quick on-the-fly updates. But what solution would you suggest in the chatbot case?

Business case 1

Problem description: a couple of new bots are about to launch on the Papua-New-Guinean market. Today is Thursday and the launch is scheduled on Monday. The client delivery team that has built the bots example corpora has informed you that they suspect there is something worrying and suboptimal in the Papua-New-Guinean bot performance. You have looked at the standard metrics you have prepared and they are not that good but not that bad either. Quick debugging did not show any results. What are your next best actions?

1. Ensure baseline is coping well enough, maybe make a baseline a bit smarter and deploy with this, then debug the Deep models in a comfort pace.
2. Try to go and invent more sophisticated metrics that will measure in what exactly the model fails and draw further conclusions on base of that.
3. Postpone the Papua-New-Guinean bots launch and debug NLP models until success.
4. Your suggestion?

Let us vote!

Business case 1 (conclusions)

Conclusion: always have the at-hand ready decent quick baseline to revert to! Be in contact with other teams in the company! Have your MLOps ready for turbulences and configure the alerts.

Business case 2. Your client delivery team asks you: “Why is the bot catching the wrong user intent?”

One of possible solutions there:

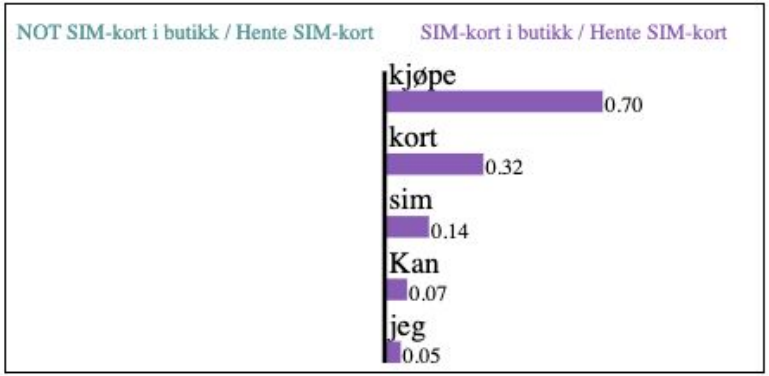
https://homes.cs.washington.edu/~marcotcr/blog/lime/ : a nice tool for explainable AI

Why did I get this response from the chatbot?

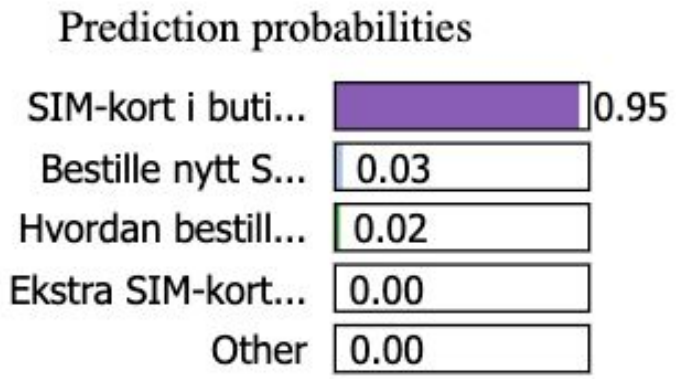
Text with highlighted words

Kan jeg kjøpe sim-kort

Word breakdown



Top predicted dialogues



Example with One Real-life Bot (Confused prediction)

- **Input**: “Jeg vil ha et abonnement med mye data”
- **Predicted dialogue**: “Datapakker og andre tjenester”

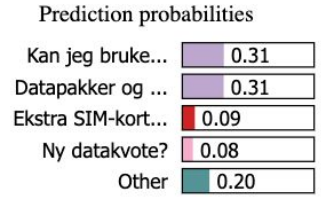
- **Correct dialogue**: “Hvilke abonnement har dere?”

Why did I get this response from the chatbot?

Text with highlighted words

Jeg vil ha et abonnement med mye data

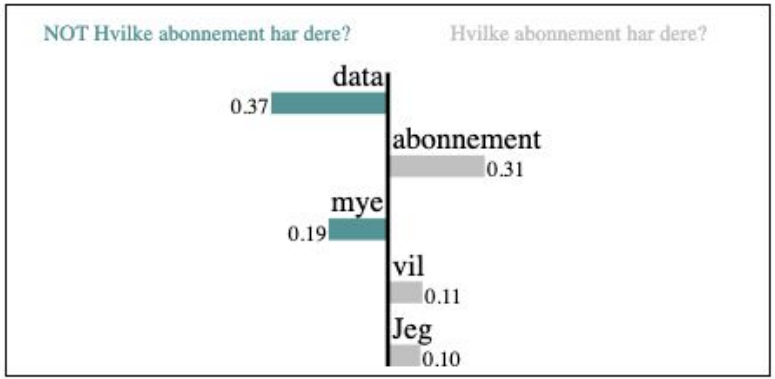
Top predicted dialogues



Wrong prediction



Ground truth



Business case 3. You have the entity expansion functionality. Your client delivery is veeeery keen on catching the specific user intent and expands the entity.

Explanation of business case

Imagine we have an entity, e.g. a car. A car can be: Volvo, Fiat, Lamborghini, Renault, Audi. The model instead of the “I want to buy a car” is being consecutively fed with: “I want to buy a Volvo”, “I want to buy a Fiat”, “I want to buy a Lamborghini”, etc.

Imagine you are feeding this examples to an LSTM. And instead of focusing only on the car brand, you are focusing also on the words “I want to buy a…”.

So “I want to buy an insurance” is being misclassified as “I want to buy a car” user intent instead of “Car insurance acquisition”.

Your suggestions to the Client Delivery team on how to feed the examples to the network differently so the issue is fixed + your suggestions on the ML team side?

Business case 4. Client asks you to delete a part of the data.

Be careful, careful, careful with how you store your data. Client privacy. Soft deleting of data (anonymizing) instead of hard delete and loosing the data.

Thank you!