Prediction is very hard, especially about conversion

Predicting user purchases from clickstream data in fashion e-commerce.

Giovanni Cassani

24th July 2019

@ JADS, Den Bosch

The views expressed in this presentation are my own and do not necessarily reflect those of Tooso, Coveo, Tilburg University and the University of Antwerp.

Who

Giovanni Cassani

Assistant Professor
(CSAI/Tilburg University)

PhD in Computational Psycholinguistics
(CLiPS/University of Antwerp)

MSc in Cognitive Science – Language and multimodal interactions
(CIMeC/University of Trento)

BA in Literature and linguistics
(University of Trento)

Jacopo Tagliabue

CTO @ Tooso

Data Scientist in various places
(Axon Vibe, ClearChannel, Diagramma , Olimpia Armani Jeans MI, iLabs)

PhD in Cognitive Sciences
(Università Vita-Salute San Raffaele, Milan)

Visiting scholar
(Computer Science and AI lab/MIT, Boston)

Start-up which provides AI, NLP e data-science solutions to improve and tailor user experience on e-commerce websites

Provider of intelligent and predictive search technologies that recently acquired Tooso

TOOSO coveo TILBURG UNIVERSITY CLiPS Computational Linguistics & Psycholinguistics University of Antwerp

Introduction

The goal

Determine as **quickly** as possible whether a user on some e-commerce website will or won't buy something using the **simplest and most general information**.

Why?

Lots of users visit websites but **leave without buying**: ample room for improvement in conversion rates!

Determine next best action based on users' intentions to maximize the likelihood they'll buy from you.

Why quickly?

You want a system that predicts what the user will do **before** the user does it. The earlier you can do it, the biggest your advantage in influencing the behavior.

Why simple and general?

Scalability: different websites may gather different details

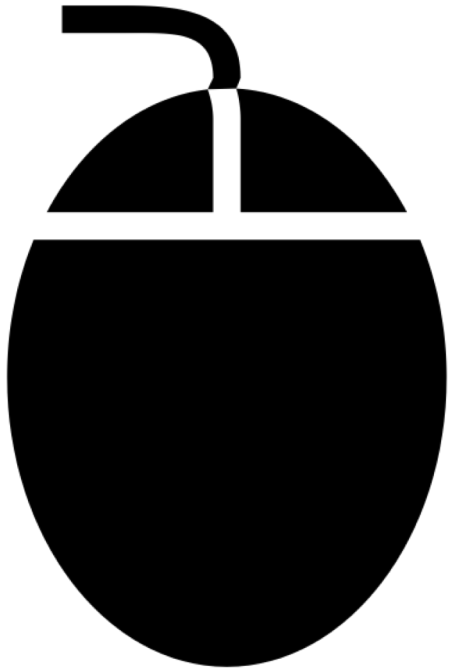
Privacy: avoid gathering and storing privacy protected data

Efficiency: need of less resources (data collection and processing)

Occam's razor: only use what you need: don't overcomplicate it!

Follow the clicks!

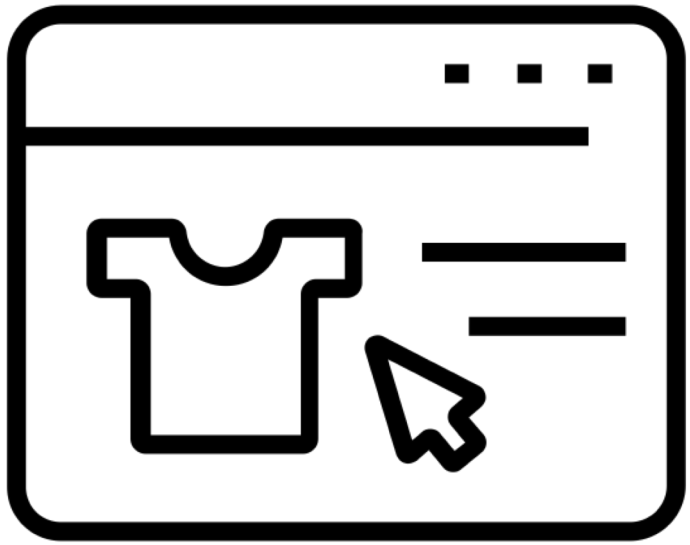
Look at where users clicks (and how long it takes them to) in order to predict their behavior



Dataset

The Tooso Fashion Clickstream dataset

- Open source
- Anonymized
- Incremental
- With time stamps
- Sessionized
  (30 minutes threshold)



Summary statistics

Total events	5,405,565
Total sessions	443,652
0/25/50/75/100th percentile for session length	1 / 1 / 4 / 12 / 7579
Event types	6 - [view, detail, add, remove, buy, click]

Dataset features

- Client-hashed ID (cookie-based)
- Session ID
- Time stamp
- Event hashed ID
- Event type
- User-hashed ID (if logged-in)
- Product hash
- Product metadata as embeddings

Dataset features

- Client-hashed ID (cookie-based)
- Session ID
- Time stamp
- Event hashed ID
- **Event type**
- User-hashed ID (if logged-in)
- Product hash
- Product metadata as embeddings

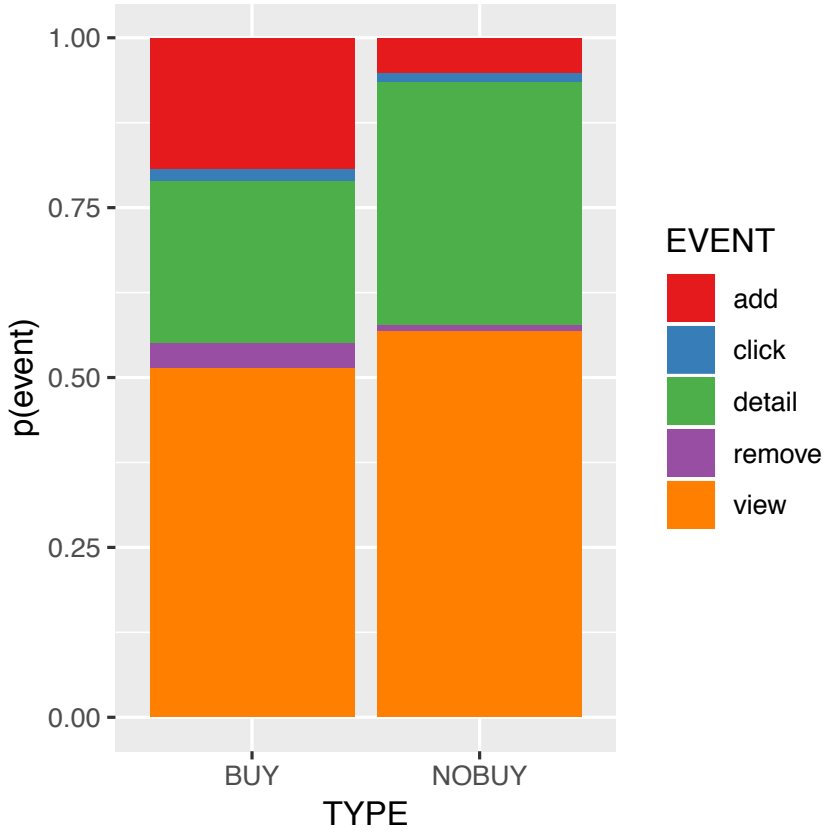
Pre-processing

Get rid of short sequences (<5 events)

Get rid of long sequences (>200 events)

Truncate BUY sequences to the last event before the first BUY event

State probabilities

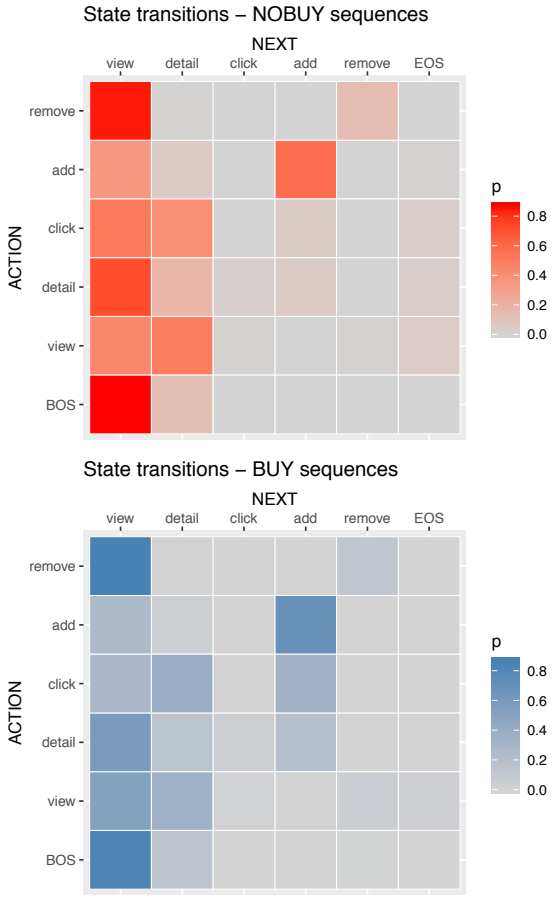


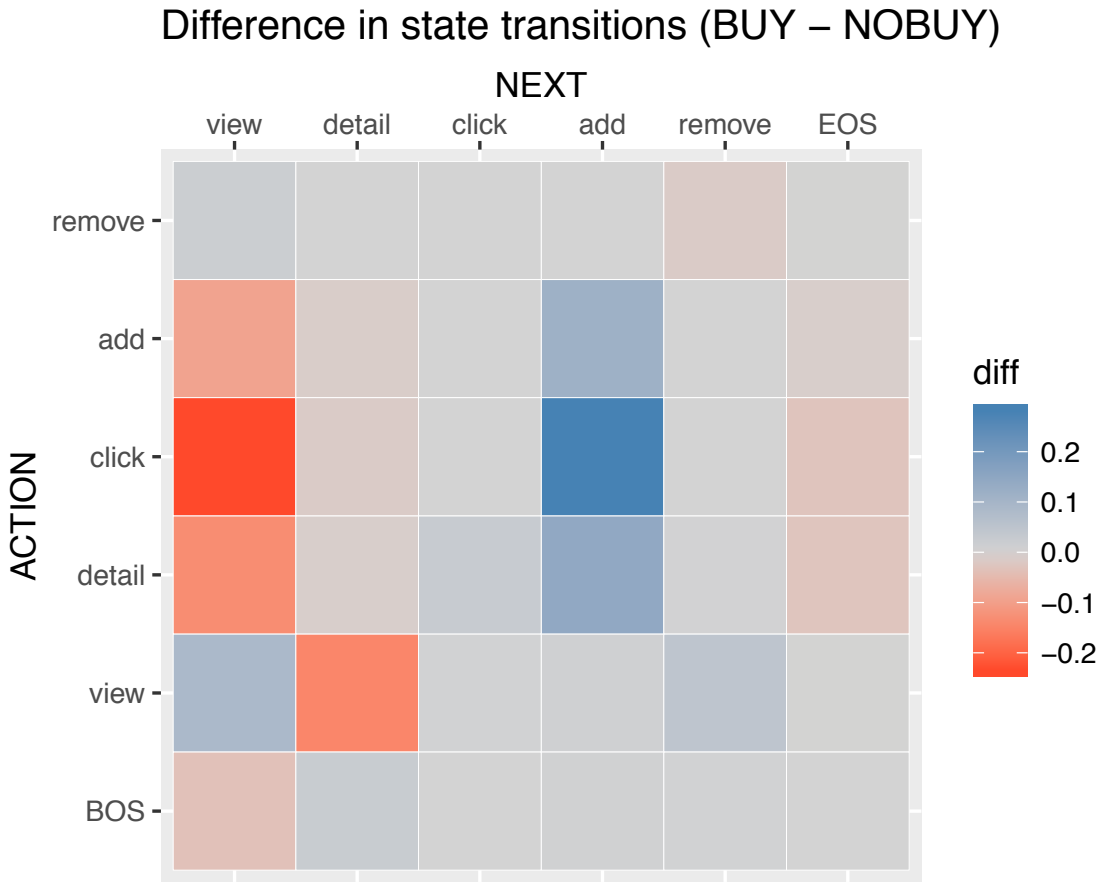
More add-to-cart and view in BUY

More detail in NO-BUY

More remove in BUY – but sparse

Sessions' transitions





Methods

Previous models

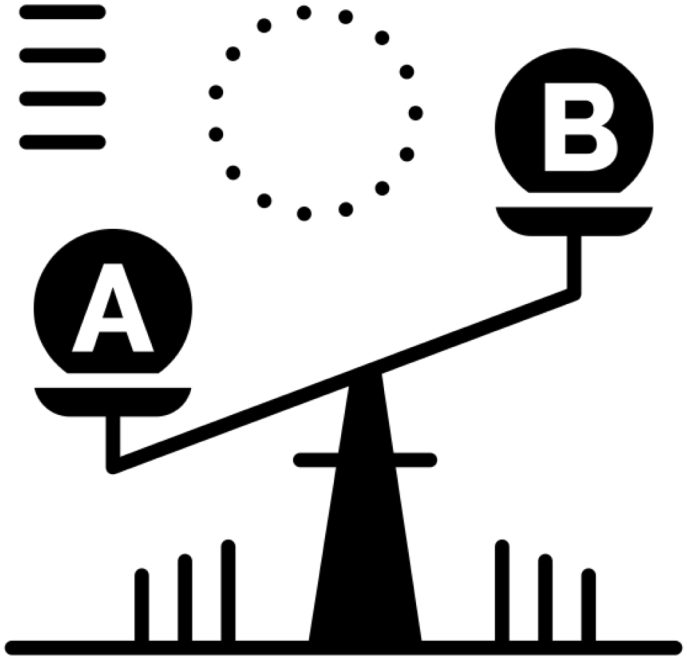
Markov Chains (+ user information)
LSTMs (with dwell times)

Main idea: the previous clicks influence the next, and different sequence classes have different transitions (~ language model).

The decision

Maximum A Posteriori

Compute the probability that the current sequence has been generated by each class sequence model and pick the class with highest probability

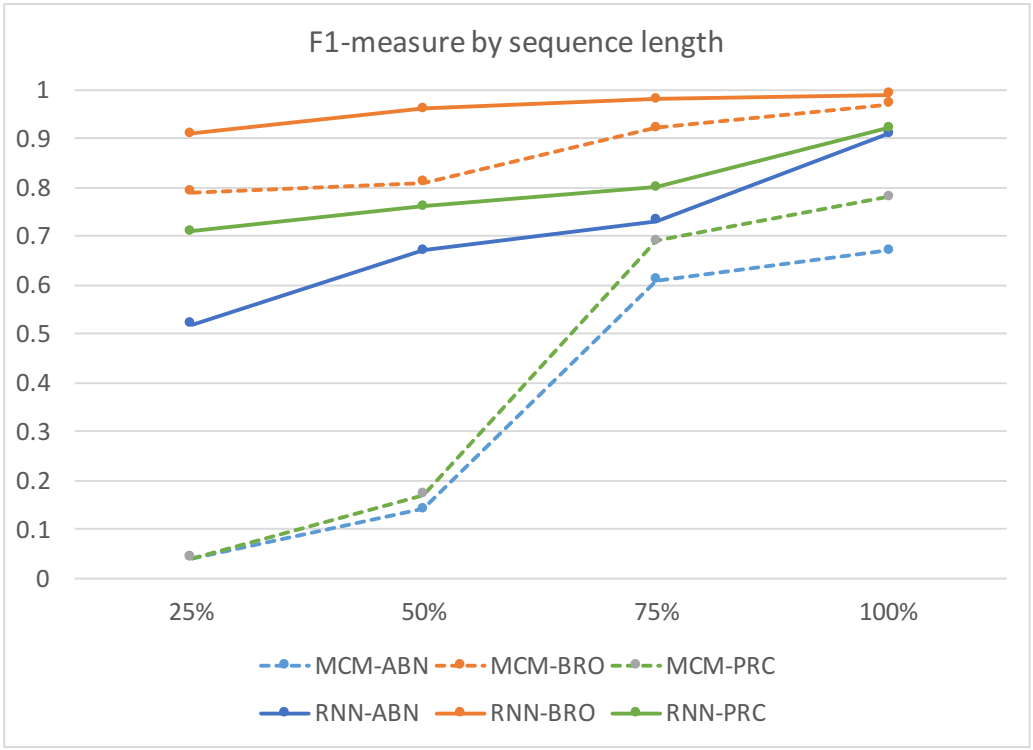


Evidence

LSTMs outperform Markov Chains.

Early classification is harder.

Plot from Toth & al [2017]



Gaps

- Not just generative classifiers
- Early event at fixed number of events, not proportion of sequence length
- Error analysis
- Insane conversation rate
  (~20% in Toth & al [2017])

Two classes

Generative vs **Discriminative**

Sequence2Label

Change the loss function of the LSTM to make it a **S2L discriminative model**.

Rather than predict the next state, take a sequence in and output a class directly.

Early prediction

Evaluate classification accuracy after **5, 6, 7, ..., 13, and 14** events

Recap

Evaluate ***generative and discriminative classifiers*** on early user intent prediction using **featurized and sessionized clickstreams** from a fashion e-commerce website.

Issues

Imbalance

Conversion rates may vary wildly across retailers.

We chose to **artificially balance** the dataset ***down-sampling*** the majority class.

Representative samples

Does it make sense to train on full sequences when the goal is early prediction?

Two training regimes

Full sequences: training happens on entire sequences

Random cropping: sequences are randomly cropped to a length between 5 and 14.

Hyper-parameter tuning

Which order for the Markov Chain? How many hidden layers and neurons for LSTMs? Which batch-size and learning rate? Which pooling for S2L models?

Grid-search

Training (70%) + validation (15%) + test (15%) split to do hyper-parameter tuning.

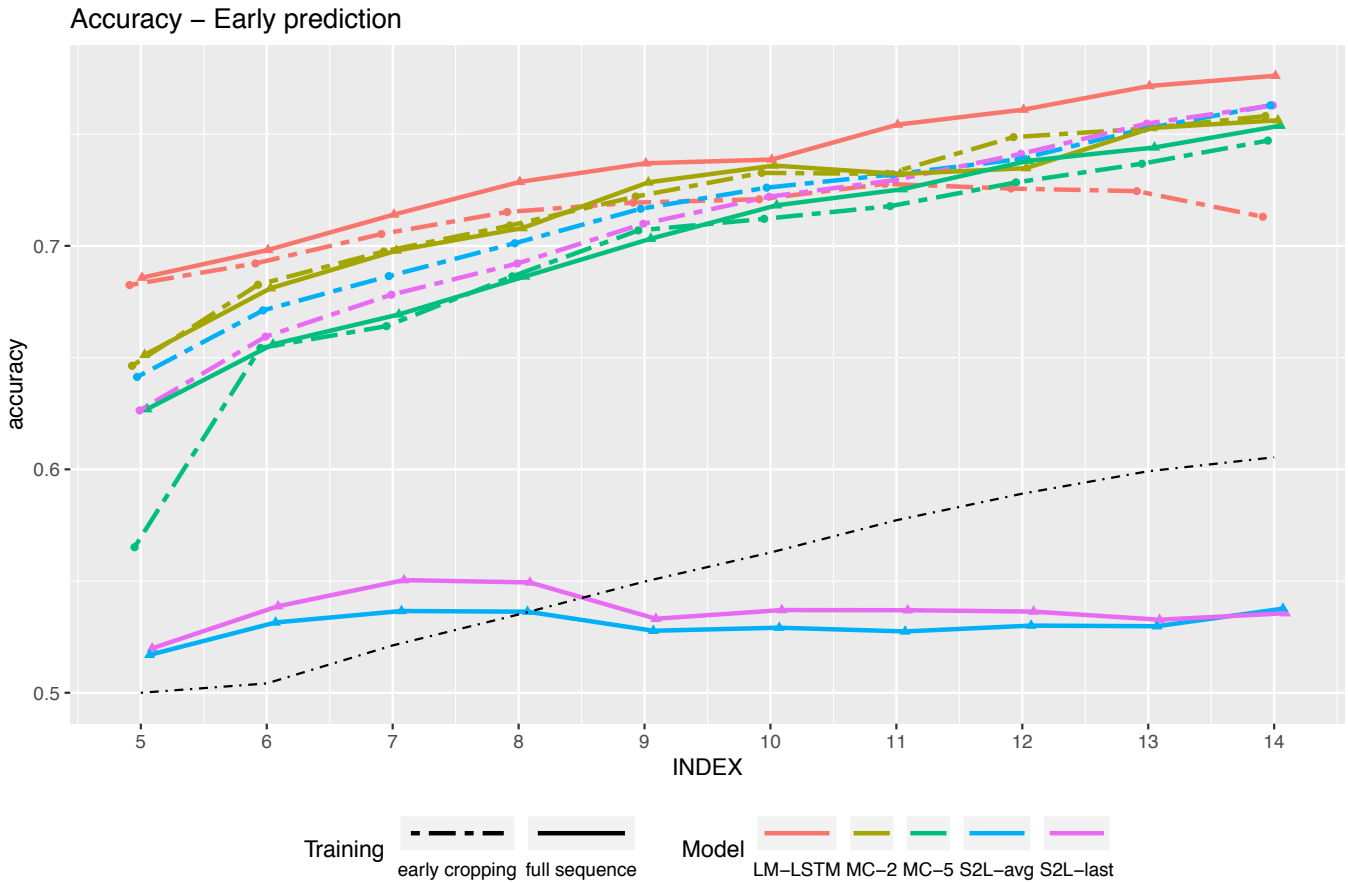
Model selection via accuracy on validation, model trained with **early stopping**.

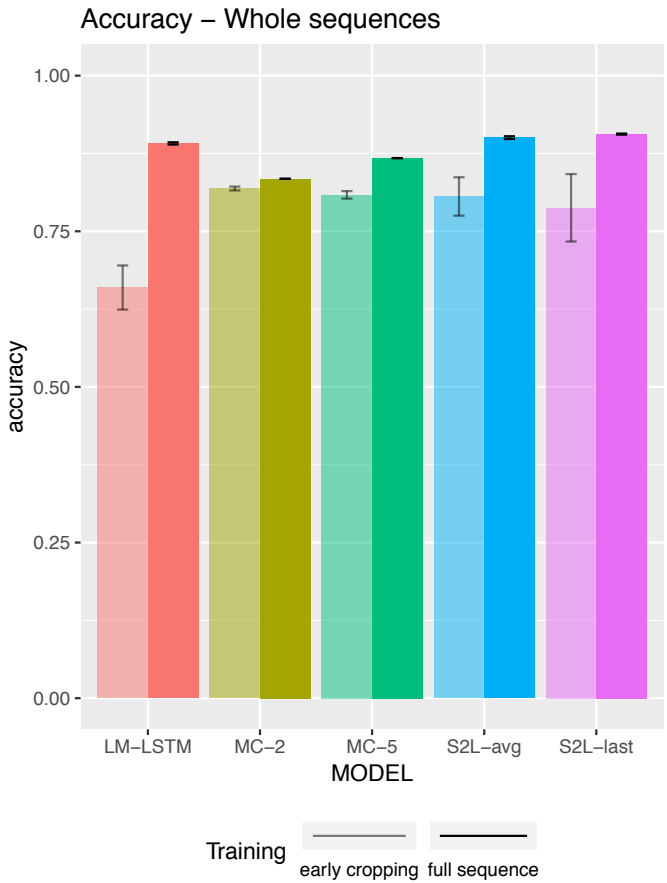
Grid-search (outcome)

Little variation across hyper-parameter constellation. In general, larger hidden layers are needed for generative LSTMs. Adding layers doesn't help.

Results

Accuracy scores



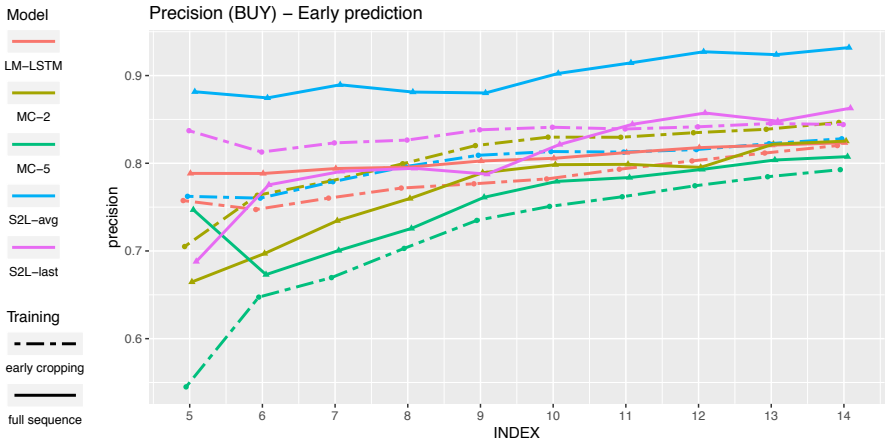


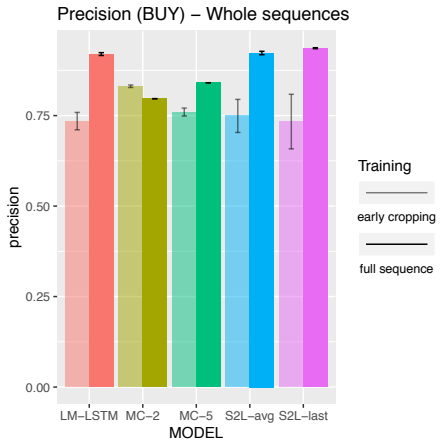
Main patterns

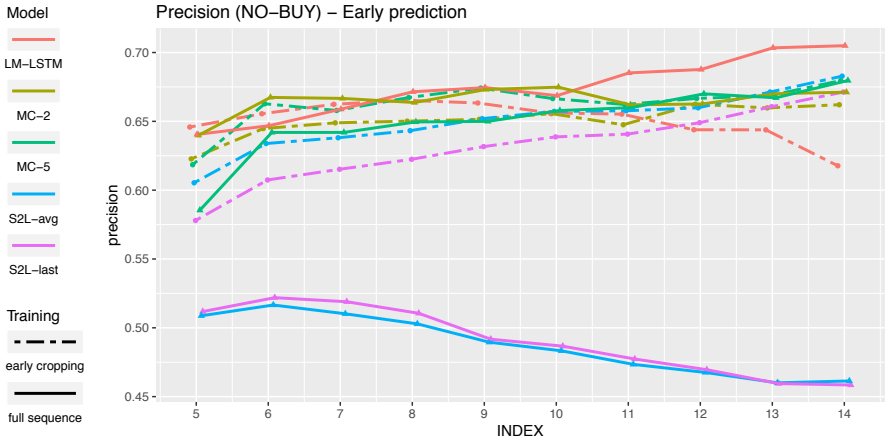
Statistical models are fairly robust but never outperform neural models.

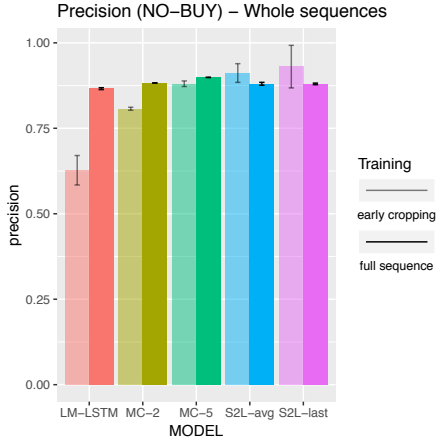
Training on cropped sequences only helps S2L models in early prediction.

Precision



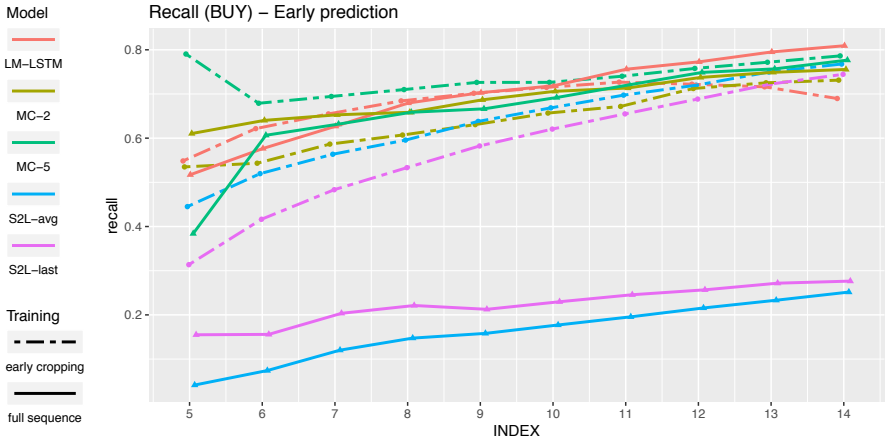


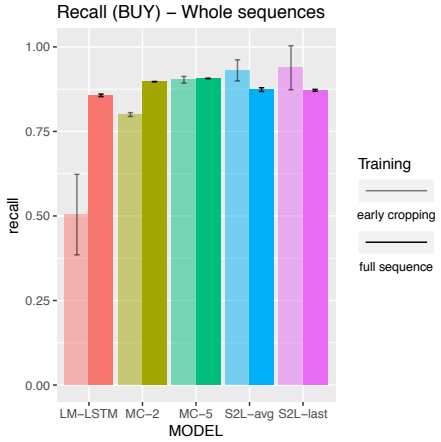


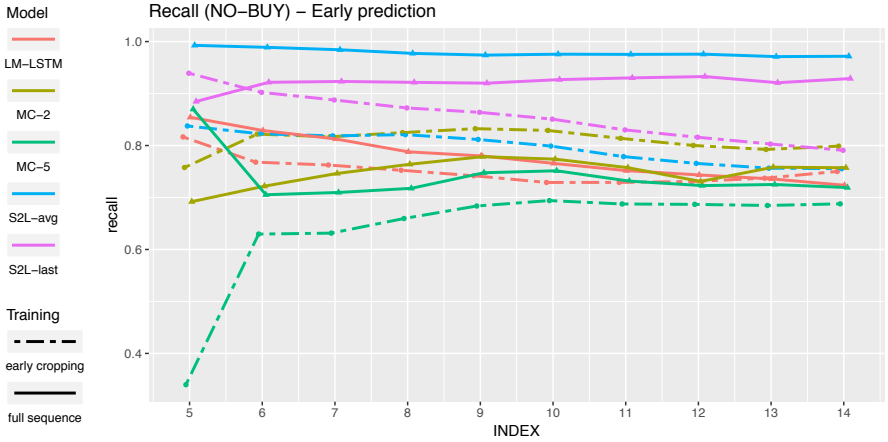


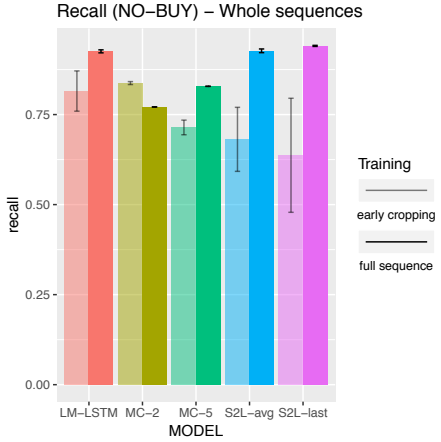
TOSO coveo Tilburg University CLiPS Computational Linguistics & Psycholinguistics University of Antwerp

Recall

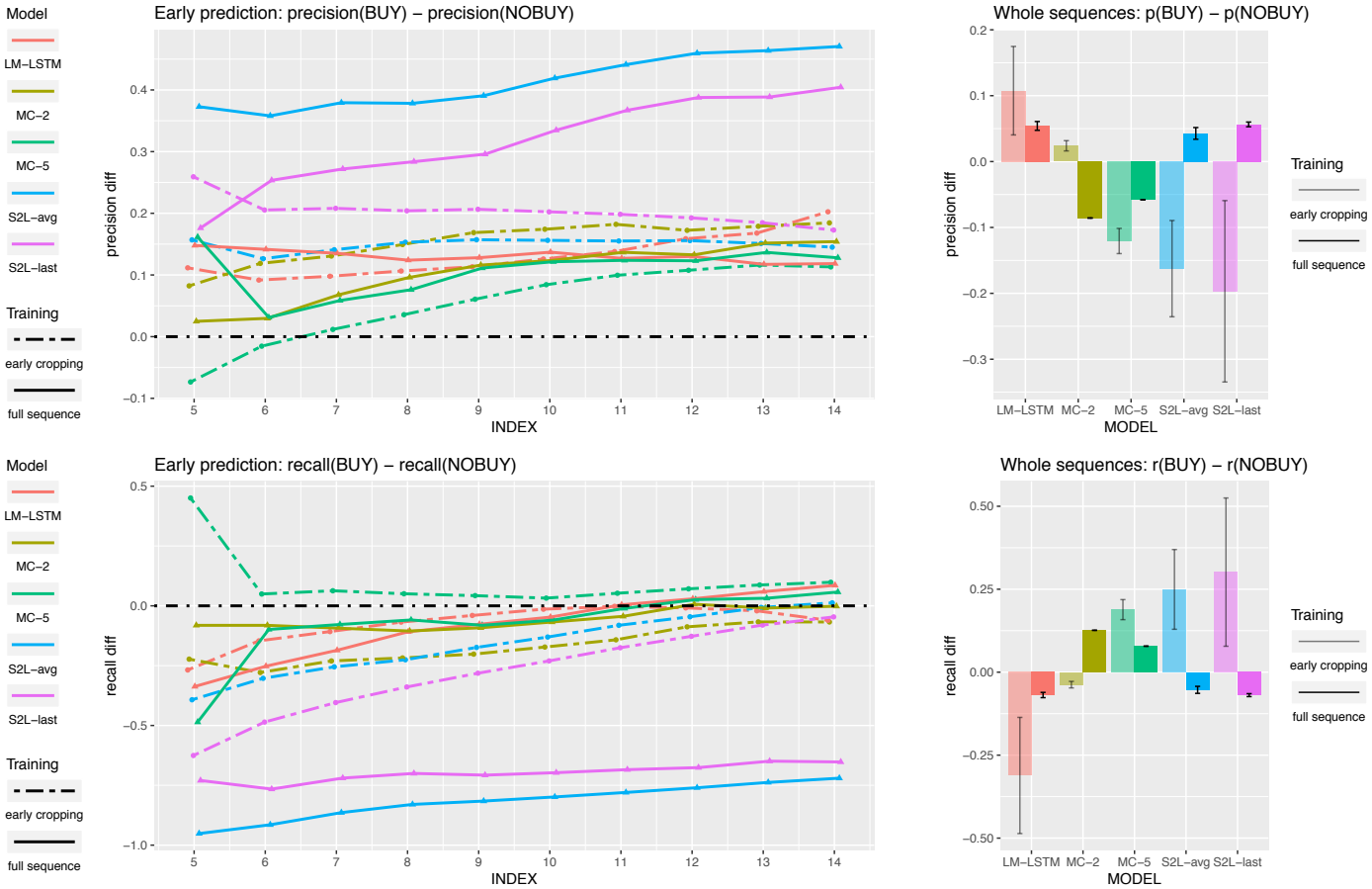

Model
LM–LSTM
MC–2
MC–5
S2L–avg
S2L–last
Training
early cropping
full sequence
Recall (BUY) – Early prediction
recall
INDEX


Recall (BUY) – Whole sequences
recall
MODEL
LM–LSTM
MC–2
MC–5
S2L–avg
S2L–last
Training
early cropping
full sequence


Model
LM–LSTM
MC–2
MC–5
S2L–avg
S2L–last
Training
early cropping
full sequence
Recall (NO–BUY) – Early prediction
recall
INDEX


Recall (NO–BUY) – Whole sequences
recall
MODEL
LM–LSTM
MC–2
MC–5
S2L–avg
S2L–last
Training
early cropping
full sequence

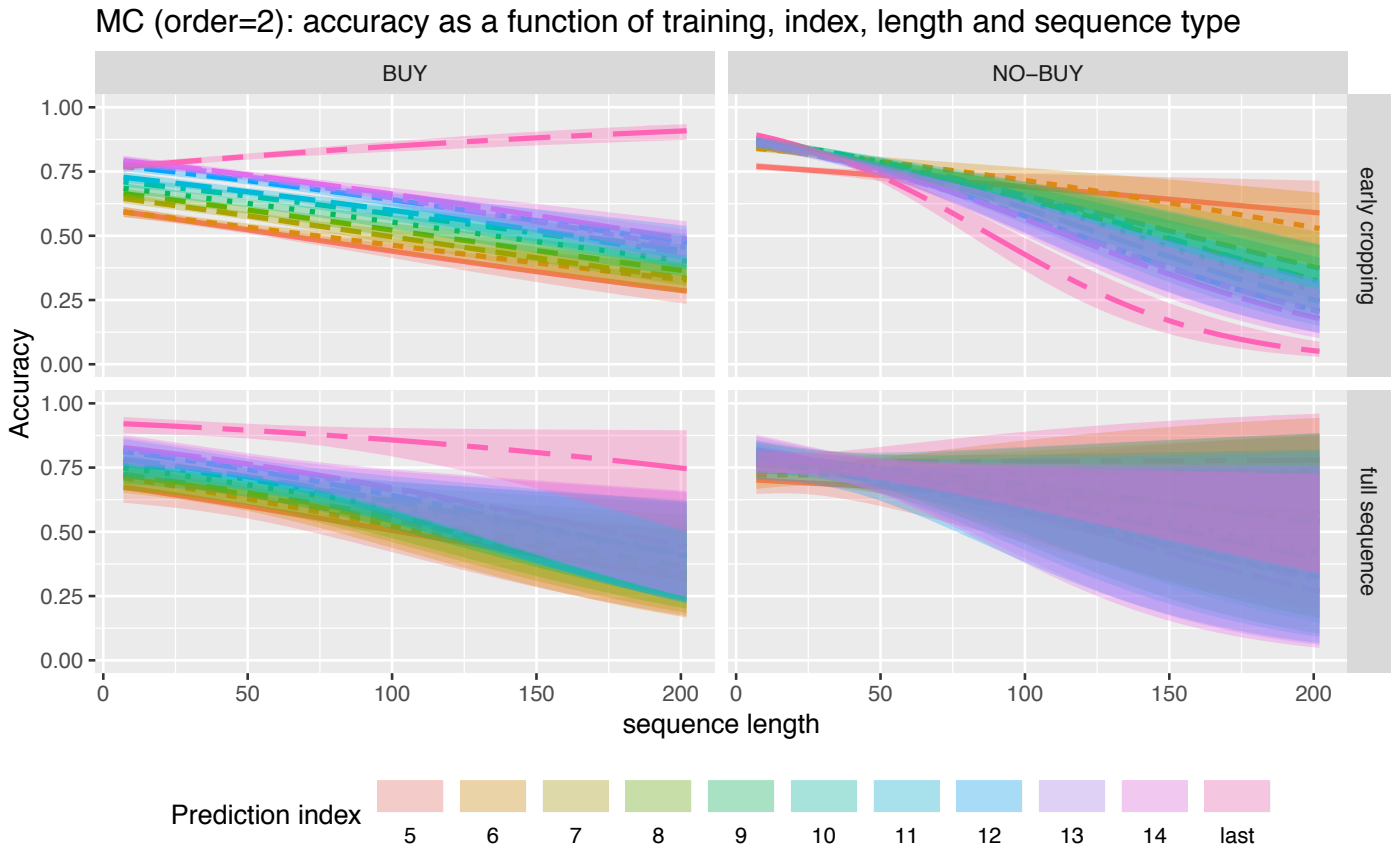
Differences by model



Slight **overextension of the NO-BUY label**: higher precision for BUY than NO-BUY sequences at early stages.

For whole sequences, cropped training flips the pattern.

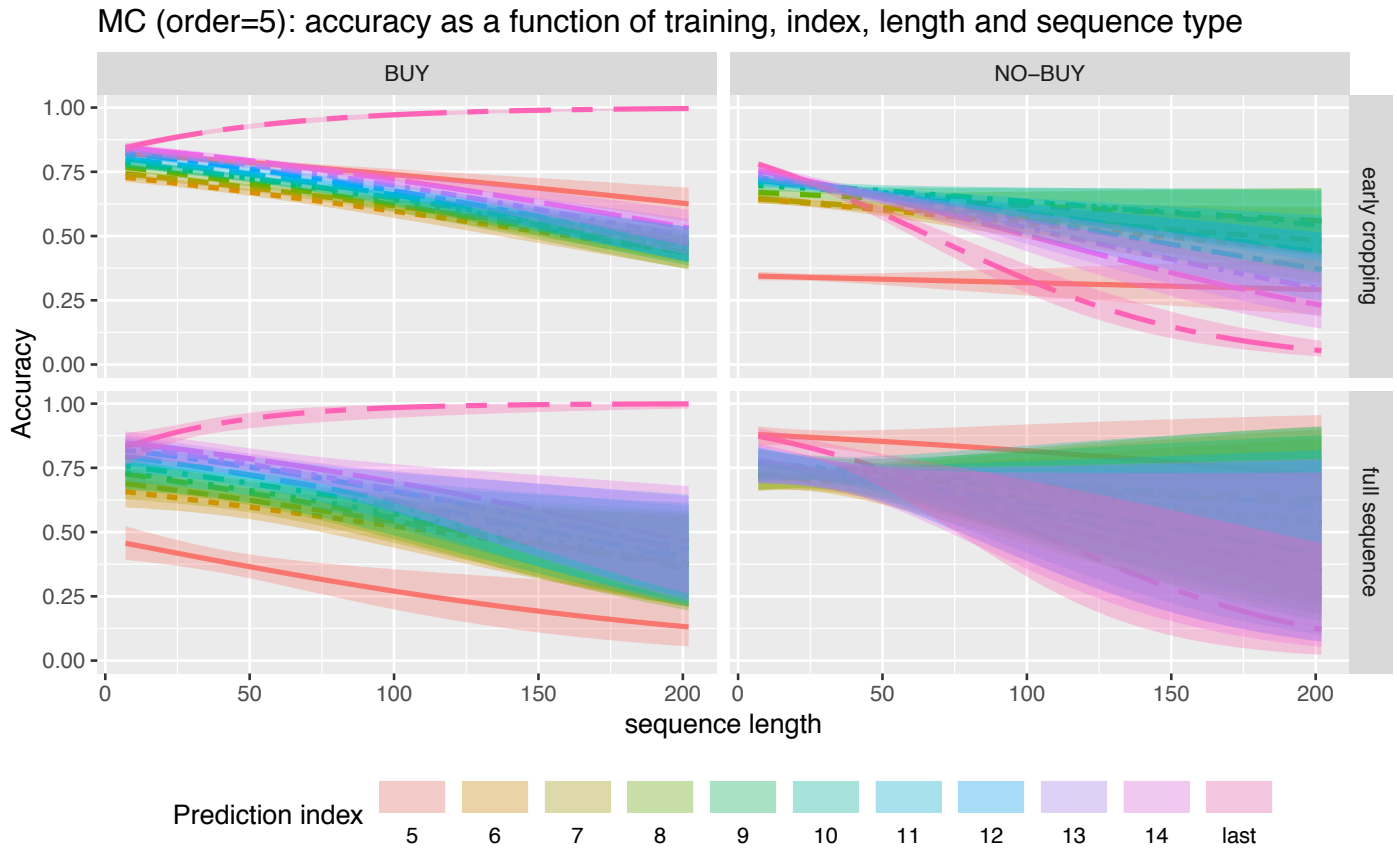
Error analysis: MC (order=2)



Generally harder to correctly predict longer sequences, except for BUY ones with cropped training.

More events help with BUY sequences, but harm with NO-BUY ones.

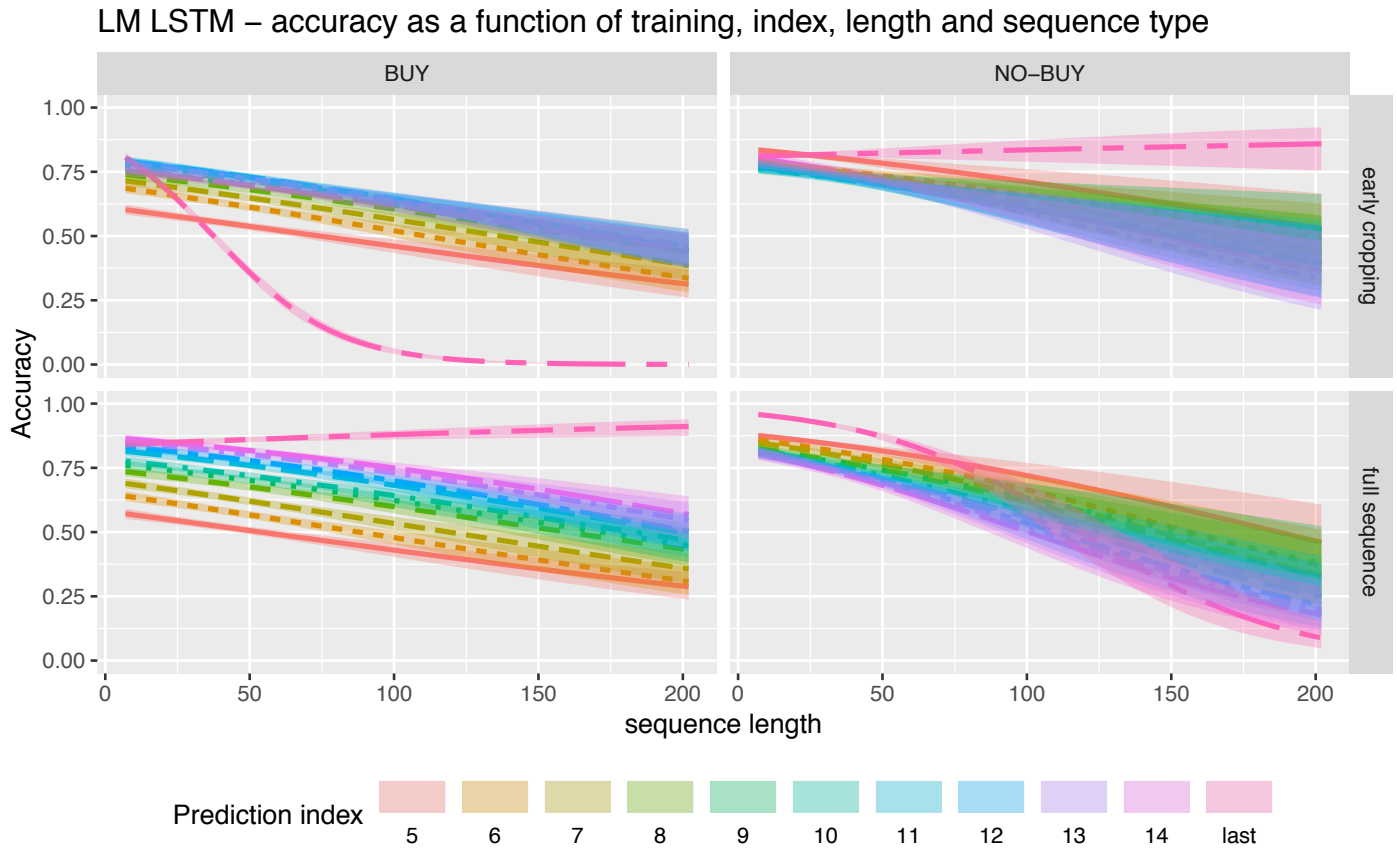
Error analysis: MC (order = 5)



Easier to predict longer BUY sequences, harder with NO-BUY ones.

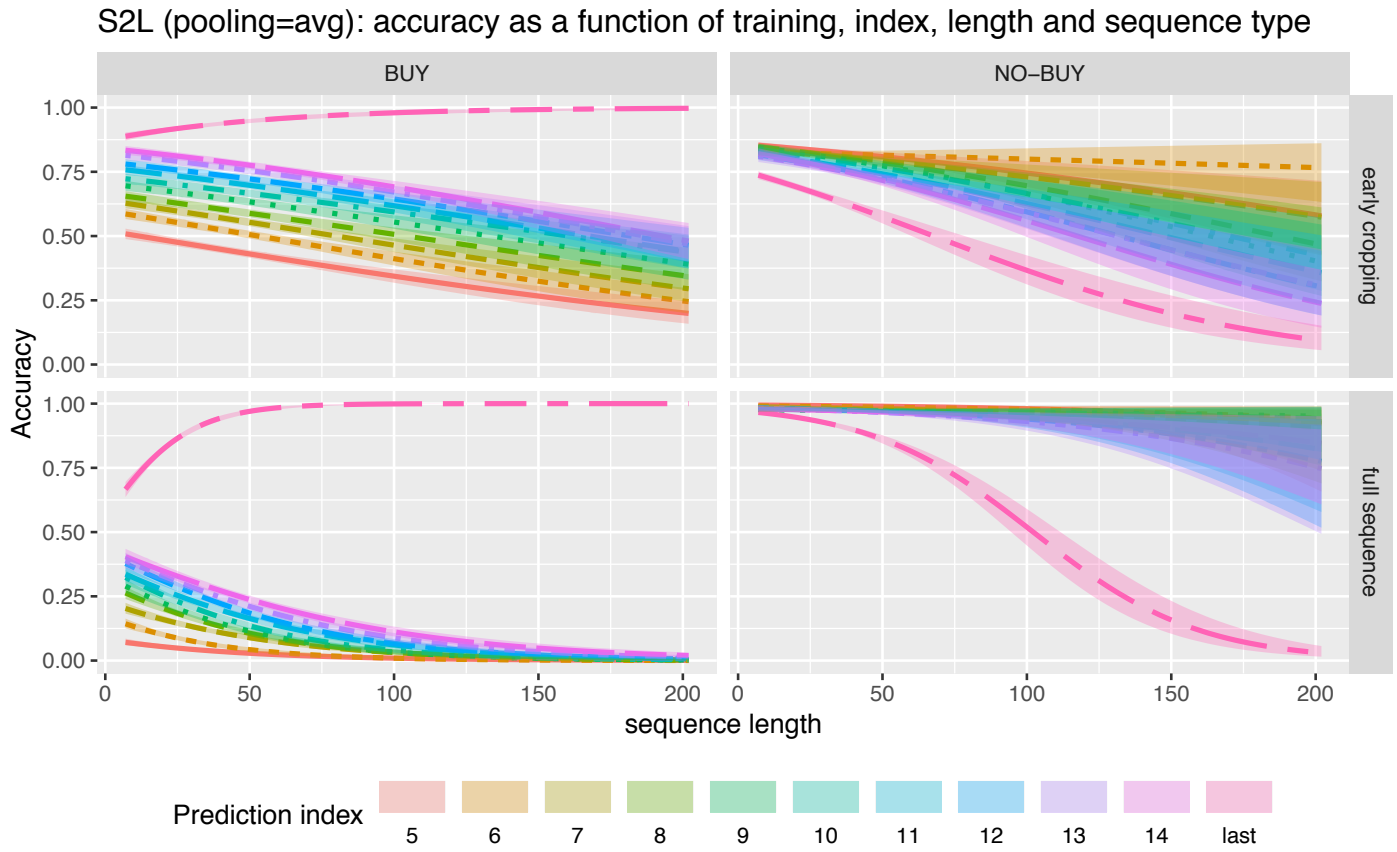
Having more clicks helps in some situations only.

Error analysis: LSTM language model



Interesting interaction between sequence type, training and length: longer sequences are classified worse when negative in full training and when positive in cropped training.

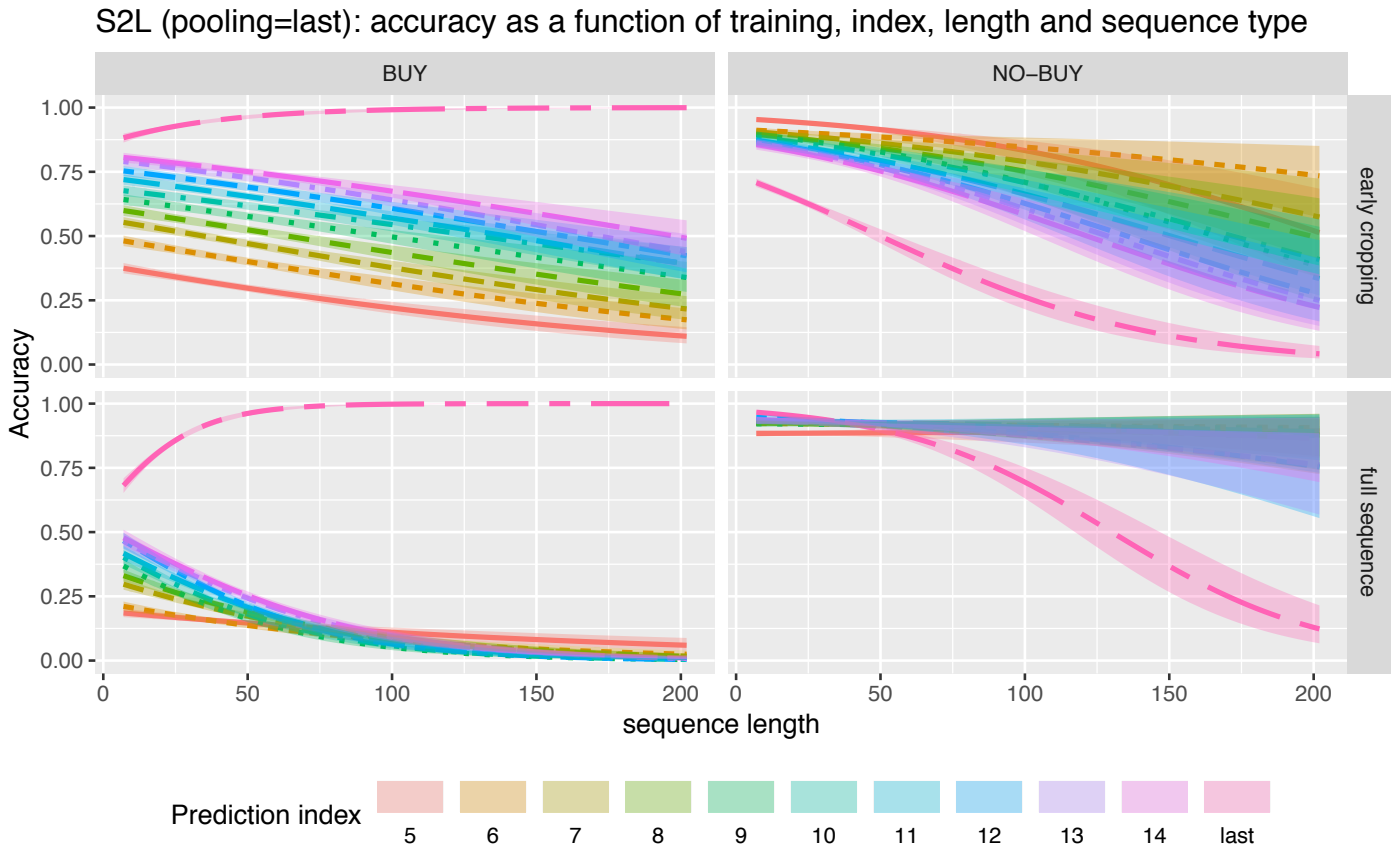
Error analysis: S2L LSTM (pooling: avg)



Longer sequences are easier when BUY but harder when NO-BUY.

Accuracy is at ceiling for shorter NO-BUY sequences when training happens on full sequences but is awful for BUY ones.

Error analysis: S2L LSTM (pooling: last)



Very similar picture to the twin model with a different pooling strategy.

More clicks help with BUY sequences, not with NO-BUY ones in the cropped training.

Conclusions

Generative v. discriminative

S2L models outperform language model-like approaches for whole sequence classification but **scale worse to online prediction**.

A grammar of clicks is useful in online prediction.

Representative data

Training on sequences that more closely resemble the test ones matters most for discriminative models.

Generative models are more robust, due to the learning goal.

Window shoppers everywhere!

All models tend to **over-apply the NO-BUY class**: models see more window shoppers than present.

Likely even worse with real imbalance in data.

More clicks help find buyers

Generally, having more clicks available for the prediction helps identifying the BUY sequences but has a negligible or even harmful effect in identifying window-shoppers.

Future steps

Understand errors

What's worse in this context? Mistake a BUY for a NO-BUY or the opposite? Why?

The upper bound

Sequences are constrained and by-class variation at early steps may be small. How far can we get with early prediction?

Improve robustness

Assess which models are **more robust wrt class imbalance** and how to train/tweak them to optimize performance across websites.

Understand robustness

Is there a relation between **within-class variability/cue validity** and class imbalance? Does this relation affect performance? How?

Promising sequences

Are there sequences that **start as** BUY but end up as NO-BUY? How many? What do they look like? What about the opposite?

Anomaly detection

Approach the problem as an anomaly detection one (spot the buyers in the ocean of window-shoppers).

More architectures

Bi-directional RNNs
Different RNNs (GRUs)
Autoencoders
Different loss functions?
Pretraining?
Suggestions?

All icons were downloaded from TheNounProject.

Authors are:

Graphic Tigers (slide 13)
Arfan Khan Kamol (slide 15)
Flatart (slide 24)

Plots were realized with ggplot2 in R.

Models were implemented in Python and run on AWS

Thank you!

Questions?