facebook

fastText

a library for efficient text classification and word representation

Piotr Bojanowski

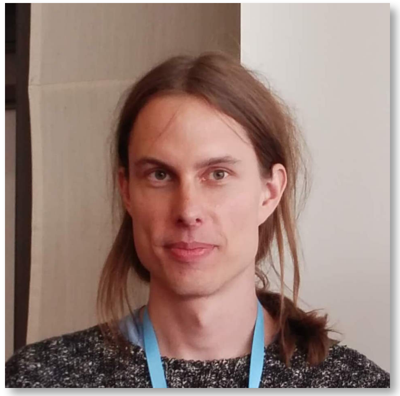
November 23th, 2016

Collaborators

Piotr Bojanowski

Edouard Grave

Armand Joulin



Tomáš Mikolov

Scientific context

- Representing words as vectors

[Mikolov et al. 2013]

Distributed Representations of Words and Phrases and their Compositionality

Efficient Estimation of Word Representations in Vector Space

- Several drawbacks:
  - No sentence representations
  - Not exploiting morphology

Taking the average pre-trained word vector is popular

But does not work very well…

Words with same radicals don’t share parameters

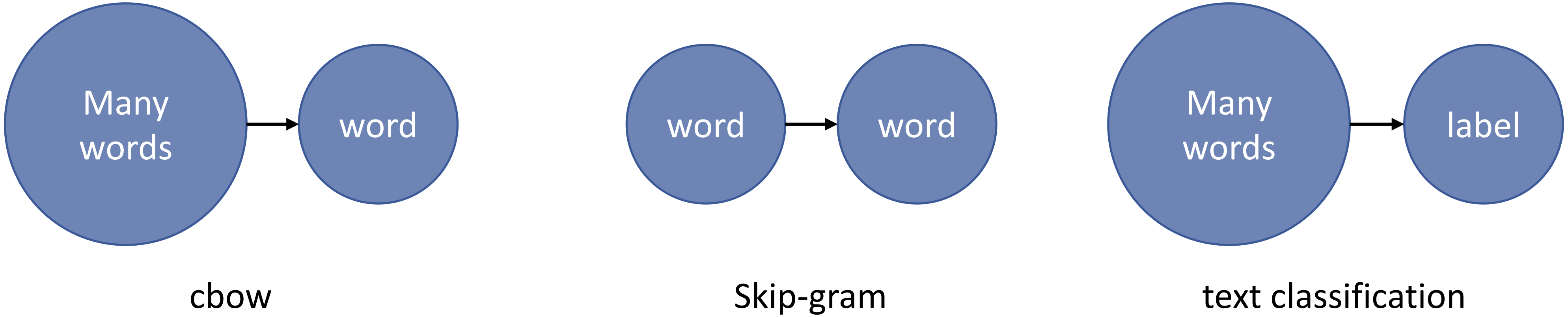
disastrous / disaster

mangera / mangerai

- Bleeding simple and fast -> widely used

Goal of the library

- Unified framework for
  1. Text representation
  2. Text classification
- Core of the library: given a set of indices –> predict an index
- cbow, skip-gram and bow text classification are instances of this model



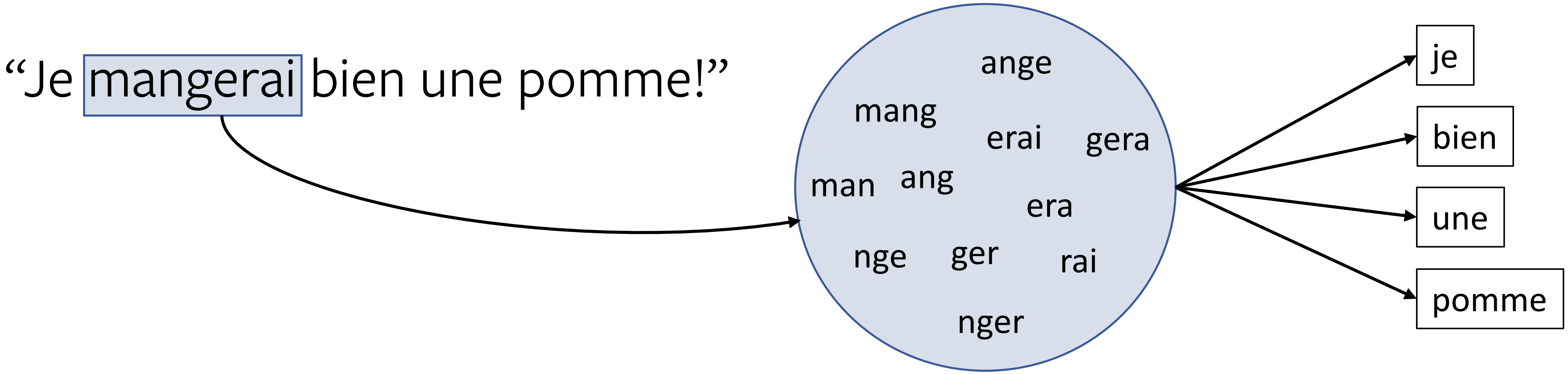
Two main applications

- Text classification

fenomeno inter is an italian sports magazine entirely dedicated to the football club football club internazionale milano . it is released on a monthly basis . it features articles posters and photos of inter players including both the first team players and the youth system kids as well as club employees . it also feature anecdotes and famous episodes from the club ' s history .

Written Work

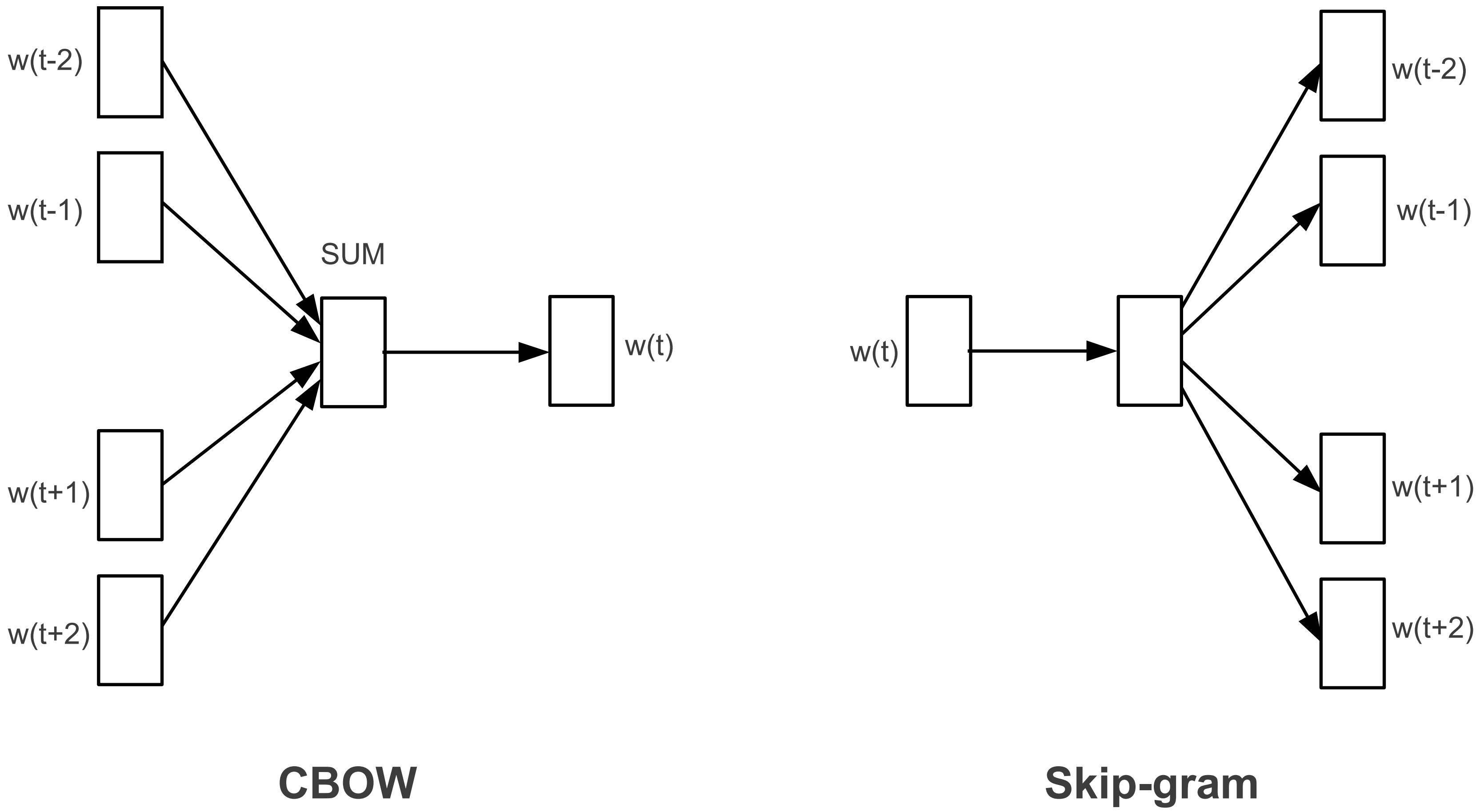
- Word representation (with character-level features)



Background knowledge

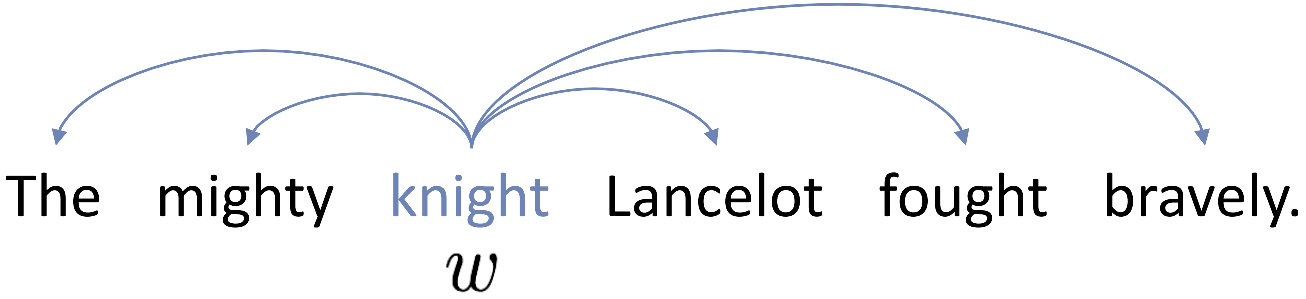
The skip-gram and cbow models of word2vec

The cbow and skipgram models



[Mikolov et al. 2013]

The skip-gram model


The mighty knight Lancelot fought bravely.

w

knight → The
knight → mighty
knight → Lancelot
knight → fought
knight → bravely.

- Model probability of a context word given a word

$$\text{feature for word } w\text{: } x_w$$
$$\text{classifier for word } c\text{: } v_c$$

$$p(c|w) = \frac{e^{x_w^\top v_c}}{\sum_{k=1}^{K} e^{x_w^\top v_k}}$$

- Word vectors $x_w \in \mathbb{R}^d$

Background: the skip-gram model

- Minimize a negative log likelihood:

$$\text{a stream of words: } (w_1, \ldots, w_t, \ldots, w_T)$$

$$\min_{x,v} \; -\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log \frac{e^{x_{w_t}^\top v_c}}{\boxed{\sum_{k=1}^{K} e^{x_{w_t}^\top v_k}}}$$

Computationally intensive!

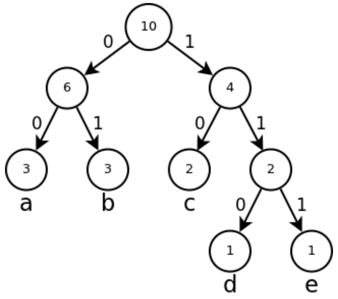
- The above sum hides co-occurrence counts

Approximations to the loss

- Replace the multiclass loss by a set of binary logistic losses
- **Negative sampling**

$$\log(1+e^{-x_{w_t}^\top v_c}) + \sum_{n\in\mathcal{N}_c} \log(1+e^{x_{w_t}^\top v_n})$$

- **Hierarchical softmax**



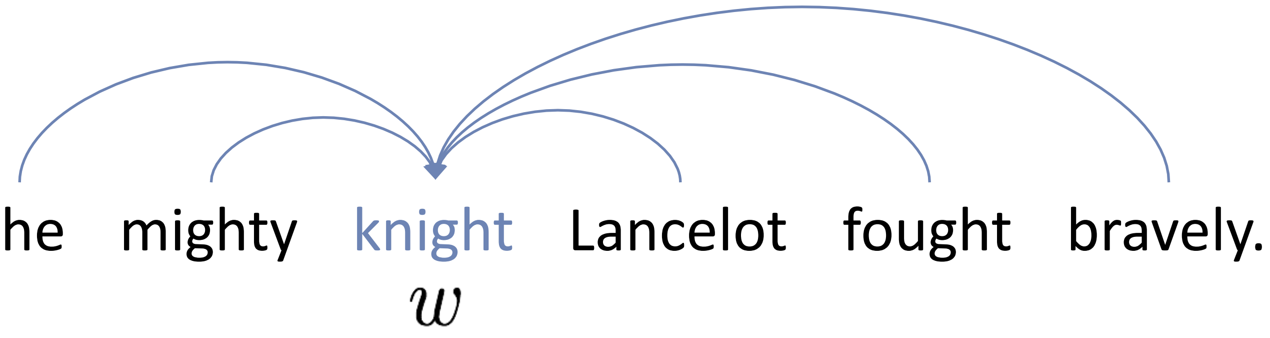
class c represented by set of codes y_{ck}

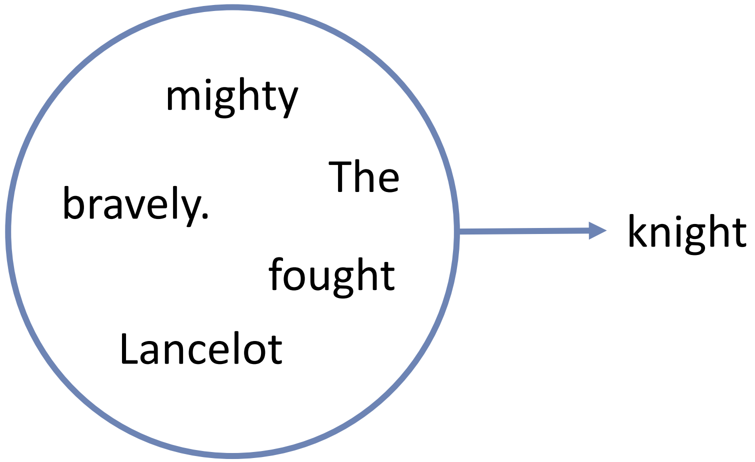
Huffman tree to generate codes

frequent classes: short codes

$$\sum_{k\in\mathcal{K}_c} \log(1+e^{y_{ck}\ x_{w_t}^\top v_k})$$

The cbow model





- Model probability of a word given a context

feature for context $\mathcal{C}$: $h_\mathcal{C}$

classifier for word w: v_w

$$p(w|\mathcal{C}) = \frac{e^{h_\mathcal{C}^\top v_w}}{\sum_{k=1}^{K} e^{h_\mathcal{C}^\top v_k}}$$

- Continuous Bag Of Words

$$h_\mathcal{C} = \sum_{c\in\mathcal{C}} x_c$$

fasttext

- Both models are instances of a broader set of models
- Different input and output dictionaries
- Common core but different pooling strategies
- Efficient and modular C++ implementation
- Allows easy building of extensions by writing own pooling

Bag of Tricks for Efficient Text Classification

Fast text classification

- BoW model on text classification and tag prediction

Starsmith (born Finlay Dow-Smith 8 July 1988 Bromley England) is a British songwriter producer remixer and DJ. He studied a classical music degree at the University of Surrey majoring in performance on saxophone. He has already received acclaim for the remixes he has created for Lady Gaga Robyn Timbaland Katy Perry Little Boots Passion Pit Paloma Faith Marina and the Diamonds and Frankmusik amongst many others.

ARTIST

Rikkavesi is a medium-sized lake in eastern Finland. At approximately 63 square kilometres (24 sq mi) it is the 66th largest lake in Finland. Rikkavesi is situated in the municipalities of Kaavi Outokumpu and Tuusniemi.Rikkavesi is 101 metres (331 ft) above the sea level. Kaavinjärvi and Rikkavesi are connected by the Kaavinkoski Canal. Ohtaans strait flows from Rikkavesi to Juojärvi.

Natural Place

- A very strong (and fast) baseline, often on-par with SOTA approaches
- Ease of use is at the core of the library

```
./fasttext supervised -input data/dbpedia.train -output data/dbpedia

./fasttext test data/dbpedia.bin data/dbpedia.test
```

Model

- Model probability of a label given a paragraph

feature for paragraph $\mathcal{P}$: $h_{\mathcal{P}}$

classifier for label l: v_l

$$p(l|\mathcal{P}) = \frac{e^{h_{\mathcal{P}}^\top v_l}}{\sum_{k=1}^K e^{h_{\mathcal{P}}^\top v_k}}$$

- Paragraph feature

$$h_{\mathcal{P}} = \sum_{w \in \mathcal{P}} x_w$$

- Word vectors are latent and not useful *per se*
- If scarce supervised data, use pre-trained word vectors

n-grams

- Possible to add higher-order features

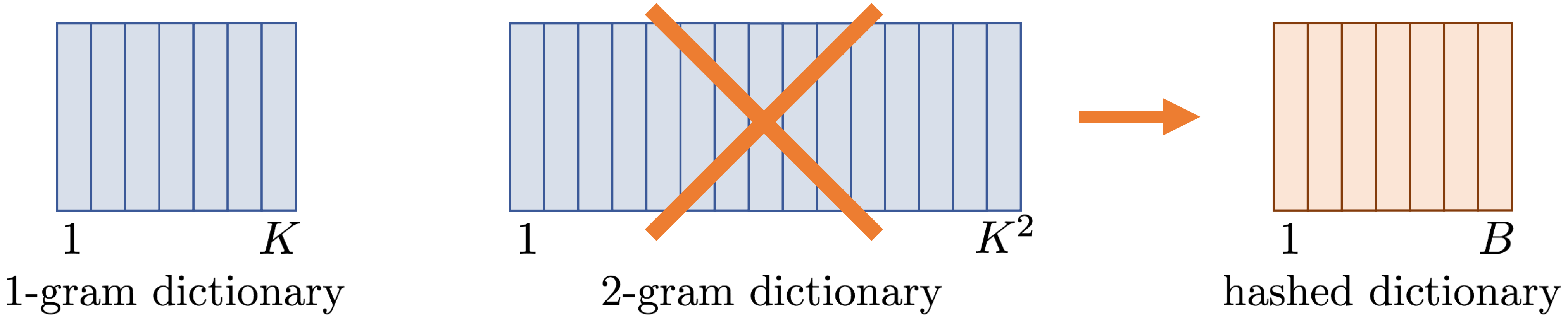
I could listen to every track every minute of every day.

I
of
to
minute
every
could
day
every
every
track
listen

every minute
track every
could listen
listen to
I could
every day
minute of
every track
of every
to every

- Avoid building n-gram dictionary

Use a hashed dictionary!

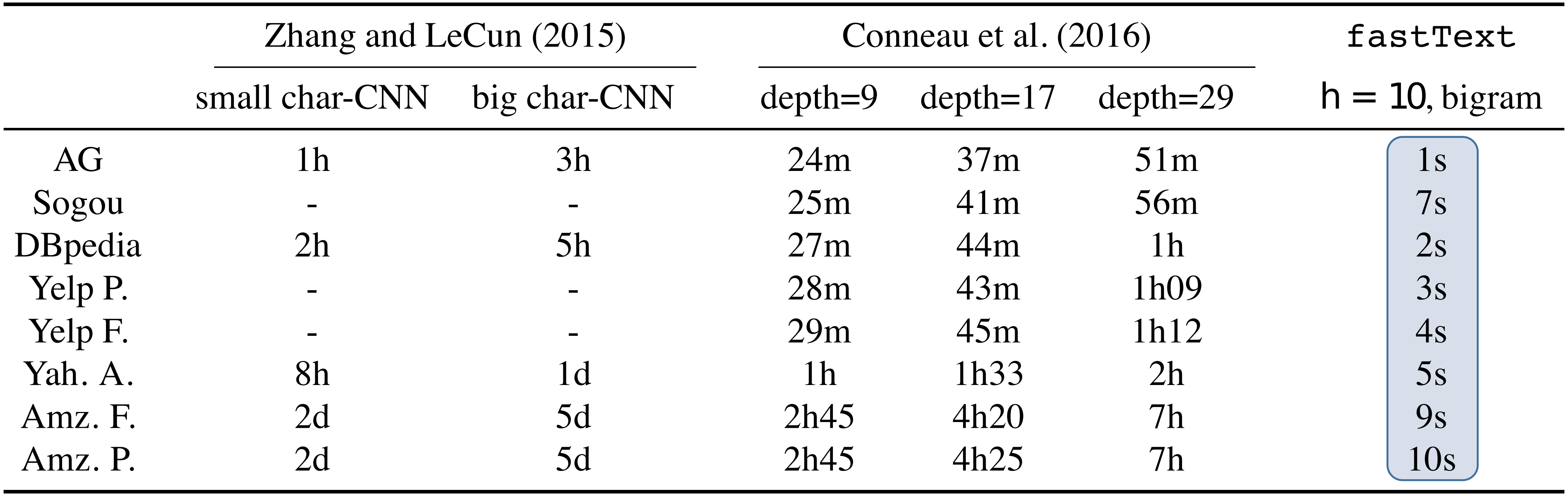


Sentiment analysis - performance

Model	AG	Sogou	DBP	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
BoW (Zhang et al., 2015)	88.8	92.9	96.6	92.2	58.0	68.9	54.6	90.4
ngrams (Zhang et al., 2015)	92.0	97.1	98.6	95.6	56.3	68.5	54.3	92.0
ngrams TFIDF (Zhang et al., 2015)	92.4	97.2	98.7	95.4	54.8	68.5	52.4	91.5
char-CNN (Zhang and LeCun, 2015)	87.2	95.1	98.3	94.7	62.0	71.2	59.5	94.5
char-CRNN (Xiao and Cho, 2016)	91.4	95.2	98.6	94.5	61.8	71.7	59.2	94.1
VDCNN (Conneau et al., 2016)	91.3	96.8	98.7	95.7	64.7	73.4	63.0	95.7
`fastText`, $h = 10$	91.5	93.9	98.1	93.8	60.4	72.0	55.8	91.2
`fastText`, $h = 10$, bigram	92.5	96.8	98.6	95.7	63.9	72.3	60.2	94.6

Table 1: Test accuracy [%] on sentiment datasets. `FastText` has been run with the same parameters for all the datasets. It has 10 hidden units and we evaluate it with and without bigrams. For char-CNN, we show the best reported numbers without data augmentation.

Sentiment analysis - runtime

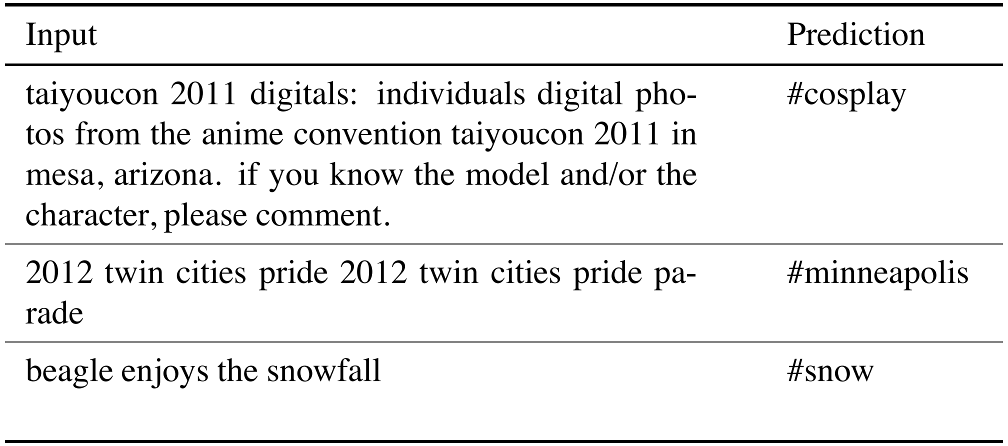


	Zhang and LeCun (2015)		Conneau et al. (2016)			`fastText`
	small char-CNN	big char-CNN	depth=9	depth=17	depth=29	$h = 10$, bigram
AG	1h	3h	24m	37m	51m	1s
Sogou	-	-	25m	41m	56m	7s
DBpedia	2h	5h	27m	44m	1h	2s
Yelp P.	-	-	28m	43m	1h09	3s
Yelp F.	-	-	29m	45m	1h12	4s
Yah. A.	8h	1d	1h	1h33	2h	5s
Amz. F.	2d	5d	2h45	4h20	7h	9s
Amz. P.	2d	5d	2h45	4h25	7h	10s

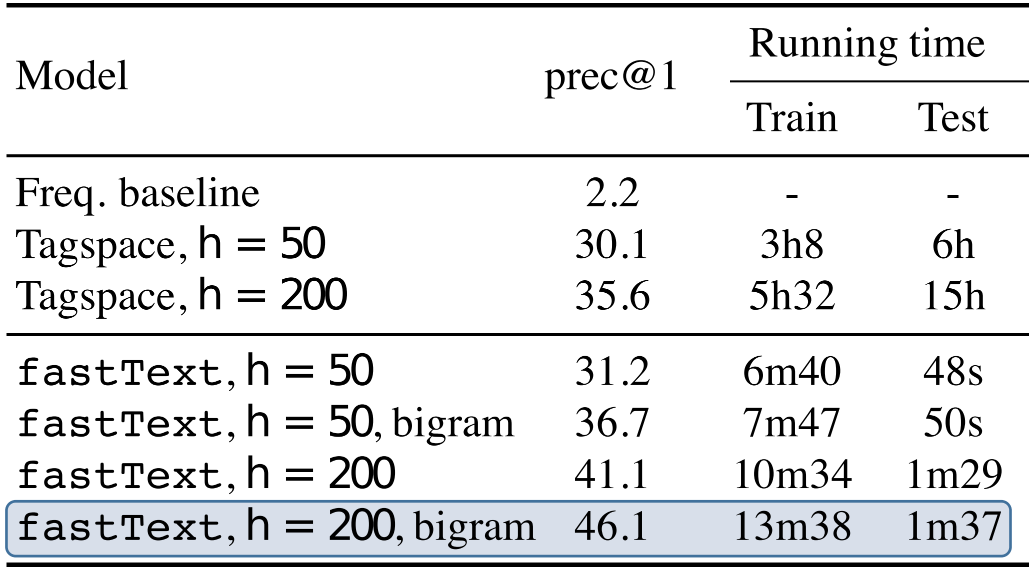
Table 2: Training time for a single epoch on sentiment analysis datasets compared to char-CNN and VDCNN.

Tag prediction

- Using Flickr Data
- Given an image caption
- Predict the most likely tag
- Sample outputs:



Input	Prediction
taiyoucon 2011 digitals: individuals digital photos from the anime convention taiyoucon 2011 in mesa, arizona. if you know the model and/or the character, please comment.	#cosplay
2012 twin cities pride 2012 twin cities pride parade	#minneapolis
beagle enjoys the snowfall	#snow



Model	prec@1	Running time	
		Train	Test
Freq. baseline	2.2	-	-
Tagspace, h = 50	30.1	3h8	6h
Tagspace, h = 200	35.6	5h32	15h
`fastText`, h = 50	31.2	6m40	48s
`fastText`, h = 50, bigram	36.7	7m47	50s
`fastText`, h = 200	41.1	10m34	1m29
`fastText`, h = 200, bigram	46.1	13m38	1m37

Table 5: Prec@1 on the test set for tag prediction on YFCC100M. We also report the training time and test time. Test time is reported for a single thread, while training uses 20 threads for both models.

Enriching Word Vectors with Sub-word Information

Exploiting sub-word information

- Represent words as sum of its character n-grams
  - We add special positional characters:
  - All ending n-grams have special meaning

`^mangerai$`

- Grammatical variations still share most of n-grams

	Singular	Plural
Nominative	uniwersytet	uniwersytety
Genetive	uniwersytetu	uniwersytetów
Dative	uniwersytetowi	uniwersytetom
Accusative	uniwersytet	uniwersytety
Instrumental	uniwersytetem	uniwersytetami
Locative	uniwersytecie	uniwersytetach
Vocative	uniwersytecie	uniwersytety

Polish declension

- Compound nouns are easy to model

Tisch Tennis Tischtennis

Model

- As in skip-gram: model probability of a context word given a word

classifier for word c: v_c

feature for word w: h_w

$$p(c|w) = \frac{e^{h_w^\top v_c}}{\sum_{k=1}^{K} e^{h_w^\top v_k}}$$

- Feature of a word computed using n-grams:

$$h_w = \sum_{g \in w} x_g$$

mang erai ange
man ang gera
era
nge ger rai nger

Character n-grams

+

mangerai

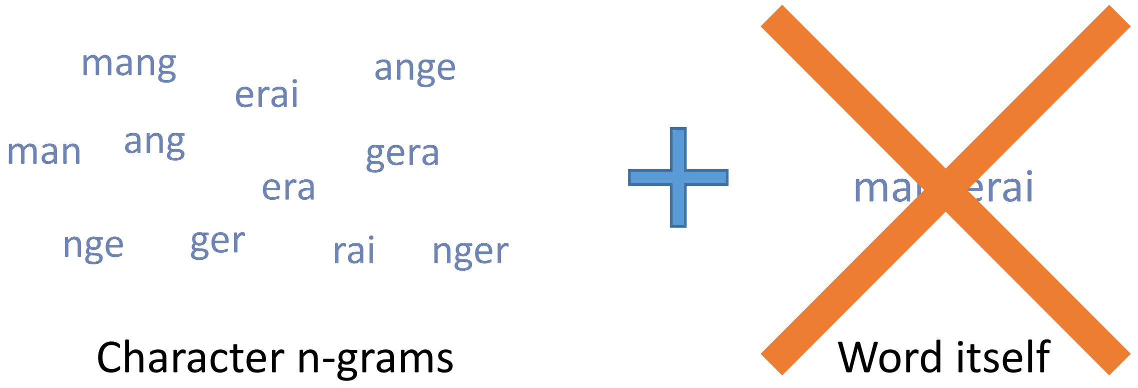
Word itself

- As for the previous model, use hashing for n-grams

OOV words

- Possible to build vectors for unseen words!

$$h_w = \sum_{g \in w} x_g$$



mang erai ange

man ang gera

era

nge ger rai nger

Character n-grams

+

ma erai

Word itself

- Evaluated in our experiments vs. word2vec

Word similarity

- Given pairs of words
- Human judgement of similarity
- Similarity given vectors

$$s(w_1, w_2) = \frac{x_{w_1}^\top x_{w_2}}{\|x_{w_1}\|_2 \ \|x_{w_2}\|_2}$$

- Spearman's rank correlation

- Works well for rare words and morphologically rich languages!

		`sg`	`cbow`	`ours*`	`ours`
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WS353	72	**73**	71	71
ES	WS353	57	58	58	**59**
FR	RG65	70	69	**75**	**75**
RO	WS353	48	52	51	**54**
RU	HJ	59	60	60	**66**

Word analogies

- Given triplets of words:

$$\text{Paris} \mapsto \text{France} \ / \ \text{Warsaw} \mapsto \ ?$$

- Predict the analogy
- Evaluated using accuracy

- Works well for syntactic analogies
- Does not degrade semantic much

		sg	cbow	ours
Cs	Semantic	25.7	27.6	27.5
	Syntactic	52.8	55.0	77.8
De	Semantic	66.5	66.8	62.3
	Syntactic	44.5	45.0	56.4
En	Semantic	78.5	78.2	77.8
	Syntactic	70.1	69.9	74.9
It	Semantic	52.3	54.7	52.3
	Syntactic	51.5	51.8	62.7

Comparison to state-of-the-art methods

	DE		EN		ES	FR
	GUR350	ZG222	WS353	RW	WS353	RG65
Luong et al. (2013)	-	-	64	34	-	-
Qiu et al. (2014)	-	-	65	33	-	-
Soricut and Och (2015)	64	22	71	42	47	67
Ours	73	43	73	48	54	69
Botha and Blunsom (2014)	56	25	39	30	28	45
Ours	66	34	54	41	49	52

Qualitative results

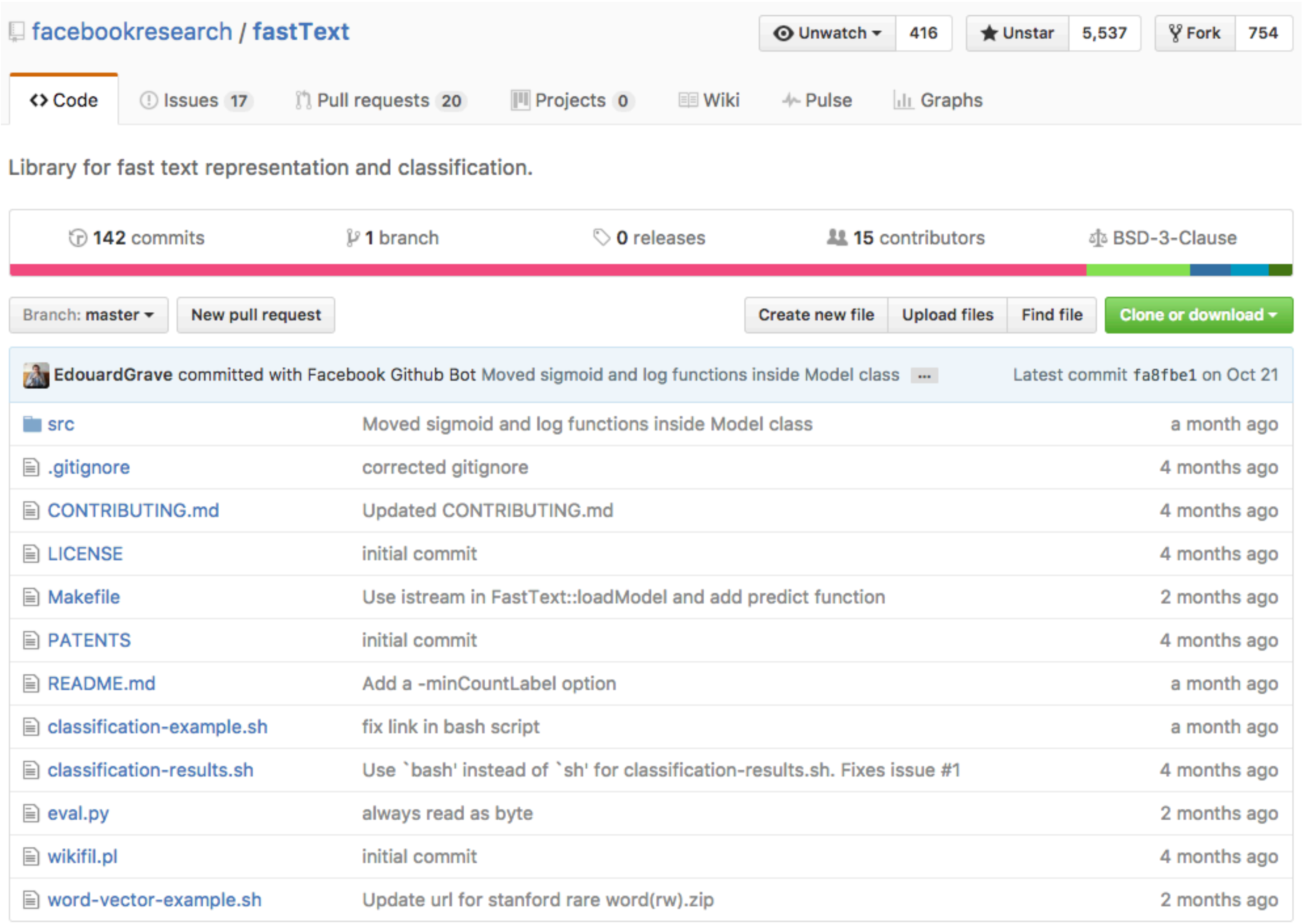
query	tiling	tech-rich	english-born	micromanaging	eateries	dendritic
`ours`	tile flooring	tech-dominated tech-heavy	british-born polish-born	micromanage micromanaged	restaurants eaterie	dendrite dendrites
`skipgram`	bookcases built-ins	technology-heavy .ixic	most-capped ex-scotland	defang internalise	restaurants delis	epithelial p53

Table 6: Nearest neighbors of rare words using our representations and `skipgram`. These hand picked examples are for illustration.

Conclusion

fasttext is open source

- Available on Github
    - After 6 months:
        - > 6700 stars!
        - 1.6k members FB group
- Featured in “popular” press
- C++ code
- Bash scripts as examples
- Very simple usage
- Several OS projects
    - Python wrapper
    - Docker files



Library for fast text representation and classification.

EdouardGrave committed with **Facebook Github Bot** Moved sigmoid and log functions inside Model class ... Latest commit `fa8fbe1` on Oct 21

src	Moved sigmoid and log functions inside Model class	a month ago
.gitignore	corrected gitignore	4 months ago
CONTRIBUTING.md	Updated CONTRIBUTING.md	4 months ago
LICENSE	initial commit	4 months ago
Makefile	Use istream in FastText::loadModel and add predict function	2 months ago
PATENTS	initial commit	4 months ago
README.md	Add a -minCountLabel option	a month ago
classification-example.sh	fix link in bash script	a month ago
classification-results.sh	Use `bash' instead of `sh' for classification-results.sh. Fixes issue #1	4 months ago
eval.py	always read as byte	2 months ago
wikifil.pl	initial commit	4 months ago
word-vector-example.sh	Update url for stanford rare word(rw).zip	2 months ago

Questions

facebook