INTEGRATION LEXICON WITH MACHINE LEARNING FOR SENTIMENT ANALYSIS

Rusnachenko N.L.[1], Loukachevitch N.V.[2]

RUSSIR 2017, Yekaterinburg, Russia

1. Problem

- Building ML model for Twitter messages **sentiment classification task**. (SentiRuEval competition)
- **Sentiment class defines** for whole message, and shows relationship between message and company mentioned in it.
- For each domain this problem resolves separately:
  - BANK – bank companies;
  - TCC – telecommunication companies.
- Each message could be labeled with one of the following scores: {1, 0, -1}

2. Approach

- **Classifier:**
  - SVM/LR
    - Embedding: tf-idf
    - Use balanced collections
  - *Neural networks*
    - RNN, GRU, LSTM
    - Embedding: w2v models
- **Extra Features:**
  - Build Lexicons (see 3.) based on Corpora
- **Handcrafted features[1]**, amount:
  - UPPERCASE words
  - signs ('?', '!', '…')
  - $\sum, min, max$ for each Lexicon

3. Lexicons

Based on **pointwise mutual information** of terms t_1, t_2:

$$\boldsymbol{PMI}(t_1, t_2) = \log_2 \frac{P(t_1 \wedge t_2)}{P(t_1) \cdot P(t_2)}$$

Introducing **marker** as a second parameter of PMI function. Possible marker values: **Excellent**, **Poor**.

Semantic orientation is a function:

$$\boldsymbol{SO}(t) = PMI(t, \textbf{Excellent}) - PMI(t, \textbf{Poor})$$

- $sgn(SO(t))$ – determines the marker type of term t;
- $|SO(t)|$ – degree of belonging.

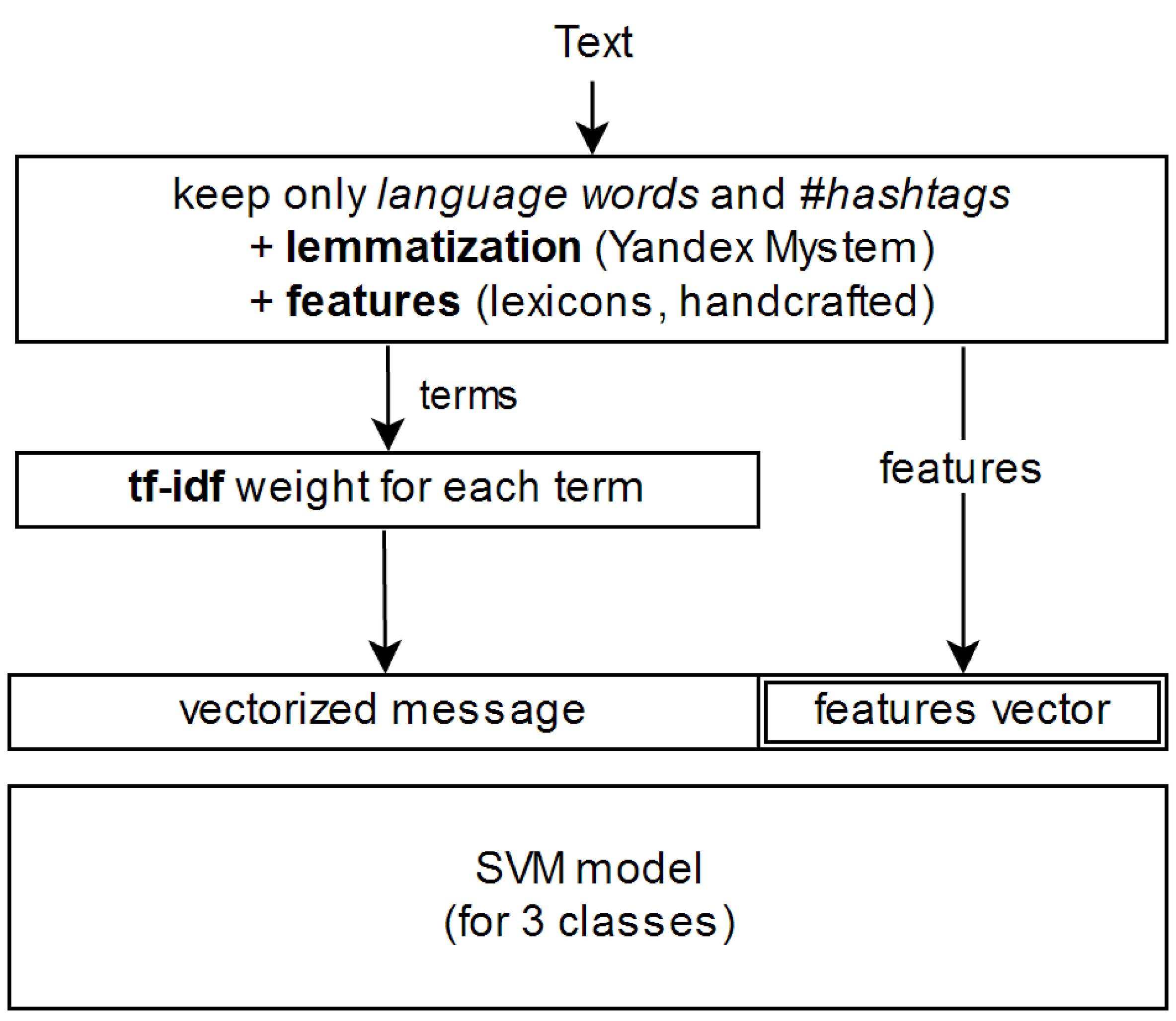
Building lexicon from messages of collection $K = K_{Excellent} \cup K_{Poor}$:

$$S: \{\, (t, SO(t)) \mid t \in K_{Excellent} \cup K_{Poor} \}$$

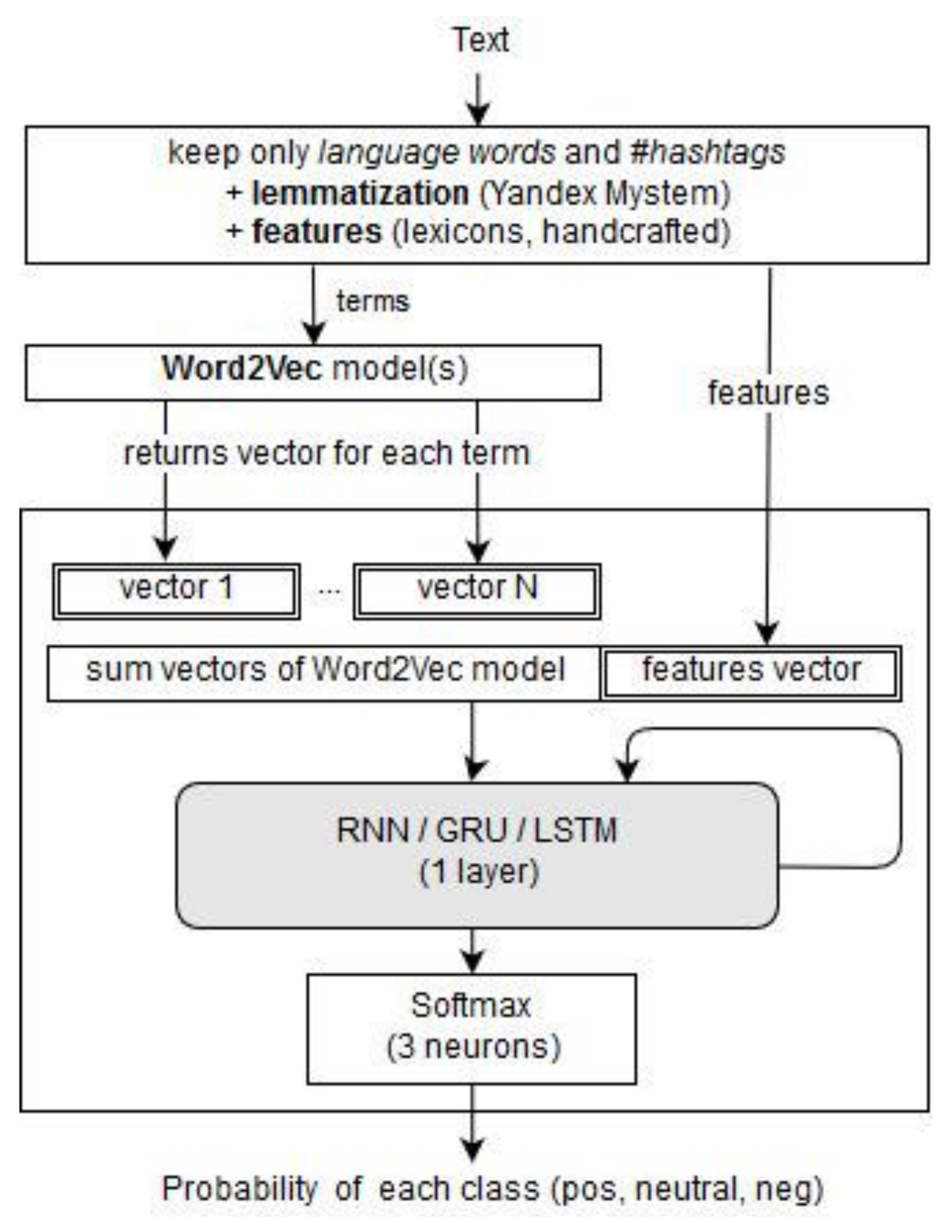
- $K_{Excellent}$ -- messages labeled **Excellent**.
- K_{Poor} -- messages labeled **Poor**.

4. SVM Model

- Implementation: $LibSVM$ [3]



5. Neural Networks



- Implementation: Theano(Python) [4]
- Optimization params:
  - X – input matrix (each row is vectorized message of train collection)
  - $rl_{curr}, rl_{def}, rl_{min}$ -- current, default, minimal values of regression coefficient respectively.
  - $grad\,(*)$ -- back propagation function
- Optimization approach (SGD):
  1. **Shuffle rows of X**
  2. Calculate *loss*
     1. $rl_{curr} := rl_{curr} * 0.5$, if *loss* greater than on previous step;
     2. Optimize otherwize (for each matrix/vector M of the model):
        $M := -rl_{curr} * grad(\text{M})$
  3. If $rl_{curr} < rl_{min}$, then $rl_{curr} := rl_{def}$.
  4. Go to next epoch

Save result after some amount of epoch. Find the best model.

6. Results

- BANK, measure: $F_{1-macro}(neg, pos)$

Model	Embedding	Training collection (B/I)*	Features	Lexicons	Result
Baseline	-	-	-	-	**18.00**
SVM	tf-idf	I	-	-	48.00
SVM	tf-idf	I	+	-	50.24
SVM	tf-idf	B	+	*all*	52.83
RNN**	$w2v, W1$	I	+	*all*	43.00
GRU**	$w2v, W2$	I	+	*all*	39.13
LSTM**	$w2v, W2$	I	+	*all*	49.00
LSTM**	$w2v, W2, W3$	I	+	*all*	51.40
LSTM	$w2v, W2, W3$	I	+	*all*	**55.32**

- TCC , measure: $F_{1-macro}(neg, pos)$

Model	Embedding	Training collection (B/I)*	Features	Lexicons	Result
baseline	-	-	-	-	**21.00**
SVM	tf-idf	I	-	-	50.90
SVM	tf-idf	I	+	-	50.69
SVM	tf-idf	B	+	*all*	**55.46**
LSTM	$w2v, W2$	I	+	*all*	50.41

* Used balanced (B) and imbalanced (I) version of training collections

** Disabling shuffle during optimization

All datasets presented in **section 7.**

7. Data & Collections Lexicons

Lexicons	Description	Terms
L_1	*Twitter* (using streaming API, jan-july 2016) (AUTO)	236 091
L_2	SentiRuLex (MANUAL)	10 668
L_3	Yu. Rubtsova short message corpus (AUTO + MANUAL)	112 814

Model	Source	Messages	Embedding size
W_1	Twitter	5 000 000	300
W_2	Twitter	10 000 000	500
W_3	banki.ru	200 000	500

Imbalanced train collections

BANK	1 354 (15%)	4 870 **(55.4%)**	2 550 (29%)	8 783
TCC	704 (7%)	6 756 **(74.22%)**	1 741 (19%)	9 102

Balanced train collections

BANK	14610
TCC	20268

References

1. Building the State-of-the-Art in Sentiment Analysis of Tweets (Saif. M. Kiritchenko S., Xiaodan Z., 2015)
2. On the Automatic Learning of Sentiment Lexicons, Human Language Technologies (Severyn A., Moshitti A., 2015)
3. Chang C.-C., Lin C.-J. (2011), LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3):27:1-27:27
4. https://github.com/nicolay-r/tone-classifier/tree/master/models/networks/theano

[1] kolyarus@yandex.ru, Bauman Moscow State Technical University, BMSTU, Moscow
[2] louknat@mail.ru, Moscow State University, Moscow