Similarity Learning with Feedback for Invoice Line Item Matching

Chandresh Kumar Maurya
Post-Doc fellow

IBM Research, Bangalore

November 22, 2018

Outline

Similarity Learning with Feedback for Invoice Line Item Matching

Procure to Pay process

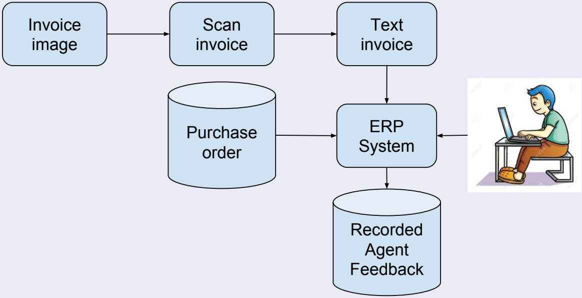


Figure: Invoice line item matching in P2P business process

The problem

Table: An example of line item matching

Invoice	PO
TRES 739mL CD KER Smooth	1. TRES 0.739L CD KER Smth 2. Tres Soya Smooth Conditioner 150 gm
5x200ml Fruit Juice 100% - Tropicana, Apple	1. Tropicana 100% Apple Juice - 1L 2. Fruit Juice 500ml - Tropicana, Custard Apple
Battery Distilled Water Replacement	1. Battery Maintenance Services 2. Battery Warranty extension

Our Contribution

- We propose two approaches to match descriptions using domain knowledge captured in the user's feedback. First approach learns similarity rank when recorded users feedback has relative ranking of description matches and second approach uses binary classification when users' recorded feedback is absolute match/no-match between pair of descriptions.
- The proposed approaches can handle OOV words based on Lexical Normalization so that STDS score does not drop due to spelling mistakes occur frequently due to OCR errors.
- We evaluate proposed approaches on real-world description datasets e.g. invoice data from internal clients, publicly available product description datasets and compare the results with the state-of-the-art approaches applied to natural language sentences.

Related Work

- In [Chechik et al., 2010], the author present an online relative similarity learning task for images. They learn a metric W based on the triples of the images within the passive-aggressive learning framework [Crammer et al., 2006]
- In [Liu et al., 2015], the author uses support vector regression with various features such as WordNet-Based features, corpus-based features, Word2Vec-based feature, Alignment-based features, and Literal-based features to predict the similarity between short English sentences.
- In [Kashyap et al., 2016] proposes a robust distributional word similarity component that combines the LSA and ML augmented data from several linguistic resources.
- In [Kutiyanawala et al., 2018, Hu et al., 2018] the author propose matching query to items in the product catalog.

Proposed Algorithm

Algorithm 1: Similarity Learning with Feedback (SLF) Algorithm

Input: Aggressiveness parameter C/learning rate η
Output: W_T

Initialize: $W_0 = I$(identity Matrix) or weight vector $\mathbf{w}_0 = \mathbf{0}$.
for $t := 1, ..., T$ **do**
- Apply Lexical Normalization as discussed in sec. 3 to the query string (e.g. Invoice string).
- Receive K strings via fuzzy matching from the pool for query string s.
- Extract noun phrases from string pair. If noun phrases did not match, return fuzzy matching score 0.
- Present the pair of strings (s, s_i) to the agent where s_i is the best fuzzy matching string.
- if the agent did not like the pair and gives negative vote, randomly sample a string s_j from the remaining pool of strings.
- if the agent prefers the pair (s, s_j) more than the pair (s, s_i), we form triple of strings (s, s_i, s_j). If the agent labels pair (s, s_i) as dissimilar and the pair (s, s_j) as similar, we form data for binary classification.
- **Update:**

$$\text{Ranking Similarty} \begin{cases} W_{t+1} = W_t + \tau_t U_t \\ \tau_t = \min(C, \ell_t^1/\|U_t\|^2) \\ U_t = [s^1(s_j - s_i)..s^d(s_j - s_i)]^T \end{cases}$$

OR

$$\text{Classification Similarity} \left\{ \mathbf{w}_{t+1} = \mathbf{w}_t - \eta \ell_t^2 \right.$$

end

Online Metric Similarity Learning

We want to learn a function $f(\cdot,\cdot)$ that assigns high score to pairs (s, s_j) than the pair (s, s_i) whenever the agent prefers (s, s_j) more than (s, s_i). Assume that the function f has a bilinear form shown in (1).

$$f_W(s_i, s_j) := s_i^T W s_j \tag{1}$$

where the matrix $W \in R^{d\times d}$. Our objective is to find the function $f(\cdot,\cdot)$ such that all the triplet strings satisfy the contraint in (2).

$$f_W(s, s_j) \geq f_W(s, s_i) + 1 \tag{2}$$

Contd...

The constraint in (2) leads to the following loss function.

$$\ell_t^1(s, s_i, s_j) = \max(0, 1 - f_W(s, s_j) + f_W(s, s_i)) \tag{3}$$

Following [Crammer et al., 2006], we can plug the above loss in passive-agressive algorithm as shown in (4).

$$\begin{aligned} W_{t+1} \quad &= \text{argmin}_W \|W - W_t\|_{fro} + C\xi \\ \text{s.t.} \quad &\ell_t^1(s, s_i, s_i) \leq \xi \quad \text{and} \quad \xi \geq 0 \end{aligned} \tag{4}$$

Datasets

Table: Summary of datasets used in the experiment

Dataset	#Train	# Test	#Features
Invoice	370	184	3649
Amazon Electronics	9368	4683	35327
Amazon Automative	21107	10553	40123
Amazon Home	21887	10943	46453
Flipkart	9417	4708	19400
SNLI	121895	60947	55956
SICK	3865	1932	18379
STS	1426	713	16110

Preprocessing

Invoice data consists of invoice strings (s). We have the corresponding PO strings (s_j) a.k.a second string) as well. Since, there is no third string (s_i) available, we manually curated and generated third string from the second one (PO string) under the following assumption so that third string is less similar to the invoice string compared to the PO string. The following rules are derived during manual curation of the invoice data:

1. Common antonyms such as men vs women.
2. Small delta Numeric addition or deletion - for second string
3. Large delta Numeric addition or deletion - for third string
4. Replace Brand names, if applicable
5. Replace Product names, if applicable
6. String Manipulation such as insertion/deletion/substitution of random character and shuffle words

Contd...

An example of string triple from invoice data look like as follows:

s: 12z Dove Men US 2in1 FRts

s_j: 11z Dove Men US 2in1 Frts

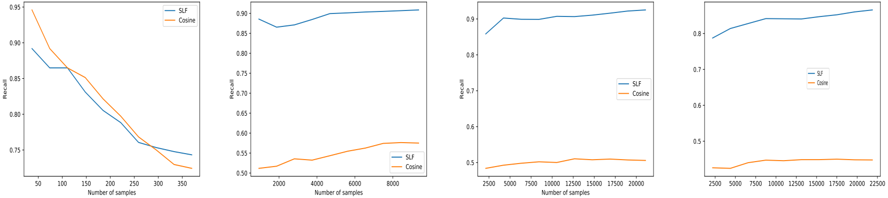
s_i: 12z Dove women US 2in1 Shampoo

Experimental Testbed and Setup

Proposed approach is compared against the following methods:

- Cosine Similarity
- Li's method [Li et al., 2006]
- UMBC [Han et al., 2013]
- DKPRo method [Bär et al., 2013]

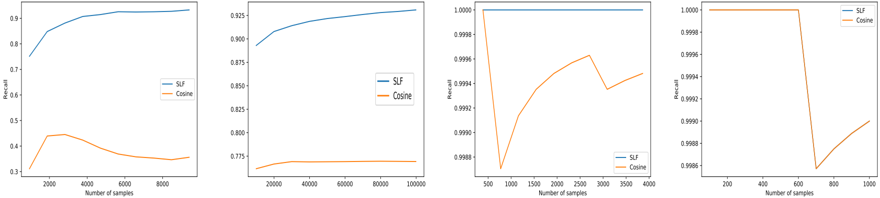
Comparative Performance Evaluation of SLFR



(a) Invoice (b) Amazon Electronics (c) Amazon Automotive (d) Amazon Home

Figure: Evaluation of online average of *recall* over various benchmark data sets. (a) invoice (b) Amazon electronics (c) Amazon Automotive (d) Amazon Home

Comparative Performance Evaluation of SLFR



(a) Flipkart (b) SNLI (c) SICK (d) STS

Figure: Evaluation of online average of *recall* over various benchmark data sets. (a) Flipkart (b) SNLI (c) SICK (d) STS

Contd...

Table: Average recall on test data

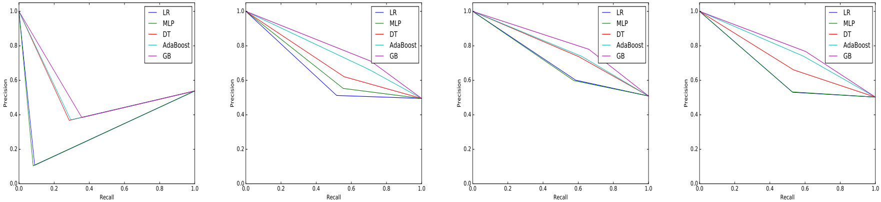
Algo	SLFR	Cosine	Li	UMBC	DKPro
Invoice	**77.71**	73.91	57.58	58.84	50.36
Amazon Electronics	**86.69**	58.27	38.27	43.30	33.67
Amazon Automotive	**90.64**	51.32	41.37	42.74	36.80
Amazon Home	**86.17**	46.11	39.93	43.29	34.91
Amazon Flipkart	**87.23**	29.33	29.37	33.0	62.96
SNLI	**94.46**	76.92	76.21	79.50	62.96
SICK	**99.97**	99.84	**99.96**	98.32	98.98
STS	99.80	**99.99**	99.86	99.0	96.47

Comparative Performance Evaluation of SLFC

Table: Recall, Precision and F-score of Gradient Boost classifier on two classes.

DataSet	Classes	Prcession	Recall	F-score
Invoie	0	0.31	0.33	0.32
	1	0.38	0.36	0.37
Amazon Electronics	0	0.72	0.72	0.72
	1	0.71	0.71	0.71
Amazon Automotive	0	0.70	0.81	0.75
	1	0.78	0.66	0.72
Amazon Home	0	0.67	0.81	0.74
	1	0.77	0.60	0.68
Flipkart	0	0.82	0.92	0.87
	1	0.91	0.79	0.85
SNLI	0	0.74	0.73	0.73
	1	0.72	0.73	0.72
SICK	0	0.69	0.69	0.69
	1	0.74	0.74	0.74
STS	0	0.78	0.57	0.66
	1	0.46	0.70	0.55

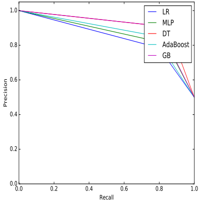
Contd...



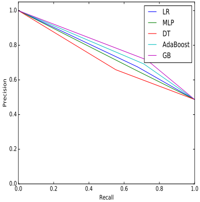
(a) Invoice (b) Amazon Electronics (c) Amazon Automotive (d) Amazon Home

Figure: Precison-Recall curve on various benchmark data sets. (a) invoice (b) Amazon electronics (c) Amazon Automotive (d) Amazon Home

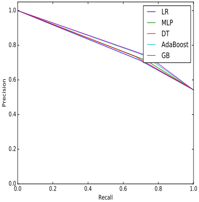
Contd...



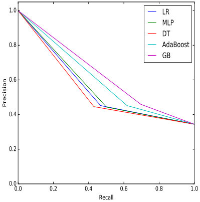
(a) Flipkart



(b) SNLI



(c) SICK



(d) STS

Figure: Precison-Recall curve on various benchmark data sets. (a) Flipkart (b) SNLI (c) SICK (d) STS

Research Problems

Problem 1 Finding Best Accommodation using Deep Aesthetic features in Multimodal Data from Social Networks

Problem 2 Recommending Popular Features for a Product based on Crowd-source Reviews of the Product and Competitor Product

Problem 3 Intelligent Reminder System using Multimodal Data

Problem 4 Generating image advertisement given a catchy tagline

Bibliography I

Bär, D., Zesch, T., and Gurevych, I. (2013).
Dkpro similarity: An open source framework for text similarity.
In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 121–126.

Chechik, G., Sharma, V., Shalit, U., and Bengio, S. (2010).
Large scale online learning of image similarity through ranking.
Journal of Machine Learning Research, 11(Mar):1109–1135.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006).
Online passive-aggressive algorithms.
Journal of Machine Learning Research, 7(Mar):551–585.

Han, L., Kashyap, A. L., Finin, T., Mayfield, J., and Weese, J. (2013).
Umbc_ebiquity-core: semantic textual similarity systems.
In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 44–52.

Hu, Y., Da, Q., Zeng, A., Yu, Y., and Xu, Y. (2018).
Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application.
arXiv preprint arXiv:1803.00710.

Bibliography II

Kashyap, A., Han, L., Yus, R., Sleeman, J., Satyapanich, T., Gandhi, S., and Finin, T. (2016).
Robust semantic text similarity using lsa, machine learning, and linguistic resources.
Language Resources and Evaluation, 50(1):125–161.

Kutiyanawala, A., Verma, P., et al. (2018).
Towards a simplified ontology for better e-commerce search.
arXiv preprint arXiv:1807.02039.

Li, Y., McLean, D., Bandar, Z. A., Crockett, K., et al. (2006).
Sentence similarity based on semantic nets and corpus statistics.
IEEE Transactions on Knowledge & Data Engineering, (8):1138–1150.

Liu, Y., Sun, C., Lin, L., Wang, X., and Zhao, Y. (2015).
Computing semantic text similarity using rich features.
In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 44–52.