# Understanding pro-social landing: prediction of funding time using loan descriptions on Kiva

**Zi Yin**
Department of Electrical Engineering
Stanford University
zyin@stanford.edu

**Yuanyuan Shen**
Graduate School of Business
Stanford University
yyshen@stanford.edu

## Abstract

Kiva is an online philanthropic crowdsourcing platform that connects lenders with the borrowers in the third world. It is via the stories conveyed by the loan descriptions that the lenders get to know the borrowers and make the decision to help. In this project we study how the loan descriptions affect the funding speed of the loans. We train recurrent neural networks and long short-term memory architectures to predict whether a loan with given descriptions and characteristics will be funded quickly. It turns out that incorporating the loan descriptions as features achieves a prominent improvement over the prediction task. We also find the words and phrases associated with higher funding rates. This is crucial for Kiva to provide guidance for the description writers and remain competitive in the market.

## 1 Introduction

Kiva is one of the world's first online philanthropic microcredit crowdsourcing platform. Its mission is to connect people through lending to alleviate poverty. Founded in 2005, Kiva has attracted more than 2 million lenders and raised more than 700 million dollars with a 98.6% repayment rate.

Kiva partnered with local microcredit lending agencies in the third world, the so-called field partners, to screen the borrowers and post their requests online. On the supply side, Kiva connects lenders online to fund the loans in increments of $25 dollars. The lenders are philanthropists who do not charge any interest for the loan. They browse through Kiva's online platform and select a fundraising loan to contribute. Each loan is listed on the website for up to a month (with some exceptions). Loans that are not fulfilled within a month will be expired. Depending on its amount, purpose (e.g. for education, for business) and description, each loan is funded at a different speed.

To remain competitive in the market, Kiva wants its loans to be funded as fast as possible. In online marketplaces where users do not see each other, loan descriptions serves as the only communication carrier between the borrowers and lenders. The loan descriptions are paragraphs ranging from 50 to 300 words in length, specifying the needs and the stories of the borrowers. Properly written loan descriptions can help the loans fully raised in hours. Hence, it is important for Kiva to understand what type of descriptions will attract more lenders (thus getting funded more quickly) and provide guidance to the partners who write the descriptions.

In this project, we study how the loan descriptions affect the funding speed of the loans using deep learning for NLP. In particular, we first classify the loans according to their funding time and then implement recurrent neural networks and long short-term memory to predict whether a loan with given descriptions and characteristics will be funded quickly. It turns out that incorporating the loan descriptions as features achieves a prominent improvement over the prediction task. We also find the words and phrases associated with higher funding rates.

## 2 Background and related work

Some previous work have also shed light on Kiva's online marketplace. Ly and Mason [1] studied the fulfilling speed of the loans by regressing the funding time of the loan on some of its quantitative characteristics. They took log transformation of the funding time and performed ordinary least square regression. They produced a model with an $R^2$ of 0.42 and suggested that loan size explains most of the variance. An increase in loan size by one standard deviation is associated with an increase in funding time by 76%. They also managed to find some other small but highly significant coefficient estimates. However, they've put too many parameters in their OLS equation and their estimates suffer from overfitting. Liu et al. [2] has studied the supply side of the market using NLP methods. They classified the lenders' self-stated motivations into ten categories with human coders (who will create the true labels) and machine learning based classifiers. They employed text classifiers using lexical features, along with social features based on lender activity information on Kiva, to predict the categories of lender motivation statements. Using the results of this classification along with Kiva teams information, they predicted lending activity from lender motivation and team affiliations. Only the category variable produced out of the NLP classifiers was selected toward their final linear regression formula while the potentially informative texts and phrases are discarded.

In the NLP literature, we have employed the long short-term memory (LSTM) architecture proposed by Hochreiter and Schmidhuber [3] and the recurrent neural network based language models in Mikolov et al. [4]. The recurrent neural network (RNN) is known to be useful for speech and translation systems when determining whether a word sequence is an accurate translation of an input sentence, while the LSTM tackles the vanishing gradient problem in standard RNNs.

## 3 The Data

We obtained Kiva's loan data from its API website. A data snapshot as of Dec 18, 2015 contains the information cumulative through that date. The Kiva dataset is of size 5.5GB on disk, in its raw form. The loan descriptions are paragraphs ranging from 50 to 300 words in length. An example:

> Tuipulotu T., 19, is single with no children. She has many years of experience in the Elei (traditional fabric printing) business. She sells to the general public 6 days per week. She has 2 previous loans with SPBD. She expects her weekly net cash flow to be 600 Tala (2̃50 USD). SPBD loans are Tuipulotu 2019s only access to capital because she was never able to qualify for a loan with the traditional banks.

Some loan descriptions are written in non-English languages and hence do not fit in our word embeddings, and some loans have missing entries. Hence, we performed a data cleaning procedure before feeding the data into the models. This procedure removes loans with missing loan descriptions. We also find that 99% of the loans in the data set are fully funded. Therefore, we remove all the loans that are not fully funded to focus on the funding speed only. Each qualified loan is transformed into a data frame, with features as the entries and the funding speed as its label. There are 860,000 loans in total after data cleaning. We then normalized the input data to have zero mean and unit variance.

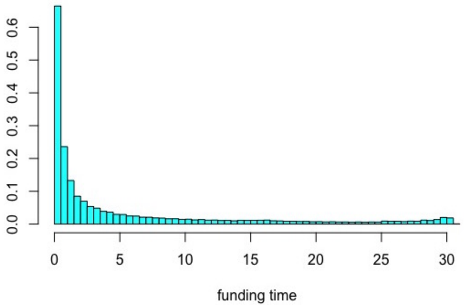

Figure 1: Distribution of funding time

| Variable | Description | Type |
|---|---|---|
| Sector | Purpose of the loan defined by Kiva | Categorical |
| Loan description | Story that explains the purpose of the loan | String |
| Loan amount | Amount that the borrower(s) requests | Numerical |
| Funded amount | Amount that has been raised on Kiva | Numerical |
| Delinquent | Whether the repayment was delayed | Binary |
| Partner ID | Id of the partner that manages the loan | Categorical |
| Posted date | Date that the loan was posted on Kiva | Time |
| Funded date | Date that the loan was fully funded on Kiva | Time |
| Disbursal date | Date that the loan is actually disbursed to the borrower(s) | Time |
| Borrower(s)'s gender | The gender of the borrower(s) | List of Binary |

Table 1: Features for each loan

Table 1 summarizes the features of the loans we are able to obtain from the data set. From the posted date and funded date, we can calculate how fast each loan was funded. The funding time, ranges from 0 to 80 days with a median of 1.4 days. Figure 1 shows the distribution of funding times we obtain from the data, from which we can see that most loans are funded within 3 days. Hence, we divide the funding time into two buckets according to the median 1.4 days. We label all loans funded within this time as fast funded loans and beyond this time slowly funded and transform our prediction task into a classification problem. The reason is two-fold. First, the regression problem, where the output has continuous range, is not robust with respect to outliers. There are times when our trained neural networks do not converge. Second, most loans are funded within a week. It is almost unlikely to accurately predict the funding times for those ”rare events”. By thresholding the loans with median, we get the relative funding performance of a loan with respect to the other loans on the platform. Eventually we want to learn about what words are associated with better performance and what words are not.

## 4 Technical Approach and Models

Our goal is to train recurrent neural network models to predict whether a loan will be fast funded or slowly funded. The word vectors for loan descriptions are our primary input. We will also include other features in Table 1 as inputs.

### 4.1 Benchmark

As a baseline measure, we run logistic regression on all the features except the loan descriptions in Table 1. For categorical variables such as partner ID, we create dummy variables to represent each category. The logistic regression model assumes that the log odds ratio is a linear function of the inputs:

$$\log \frac{P(\text{Fast funded})}{P(\text{Slowly funded})} = \langle \vec{\theta}, \vec{f} \rangle,$$

where $\vec{f}$ includes all the quantitative features in the data set and $\vec{\theta}$ is the vector of parameters. We find the maximum likelihood estimators of the parameters $\vec{\theta}$ and make predictions for all the loans on their respective probability of belonging to the fast funded class. If the probability is larger than 0.5, we predict that they belong to the faster class. Doing so, we are able to achieve an accuracy of 61.6%, which is 10% higher than random guessing.

### 4.2 Models

We build 2-layer recurrent neural networks and two-layer LSTM networks. The numbers of nodes in each layer are 256 and 144 respectively, empirically chosen by balancing training speed and accuracy. At the projection layer, we also input other features $\vec{f}$ together with the output of the underlying neural network, and produce a final softmax vector $\hat{y}_t$. Define $h_1, h_2$ to be the hidden layers, $W^{h_1 h_1}, W^{h_2 h_2}$ the recurrence matrices and $H_1, H_2$ the weights. The breakdown of the model is as follows:

$$h_1^{(t)} = ReLU(h_1^{(t-1)} W^{h_1 h_1} + xH_1 + b_1),$$
$$h_2^{(t)} = ReLU(h_2^{(t-1)} W^{h_2 h_2} + h_1^{(t)} H_2 + b_2)$$

At the projection layer, the vector of other features $\vec{f}$ is concatenated with the 2nd hidden layer

$$y_1^{(t)} = Softmax([h_2^{(t)}, \vec{f}]W + b_3)$$

To train the model, we feed the loan descriptions into the recurrent neural network. At each unfolded time point, the output $\hat{y}_i^{(t)}$ is compared to the true label $y_i$. We then use cross entropy loss and sum over all words in the loan description:

$$\sum_{t=1}^{T_i} CE(\hat{y}_t^{(i)}, y^{(i)})$$

We then sum over the training set and get the loss function we try to minimize:

$$\textbf{Loss} = \sum_i \sum_{t=1}^{T_i} CE(\hat{y}_t^{(i)}, y^{(i)})$$

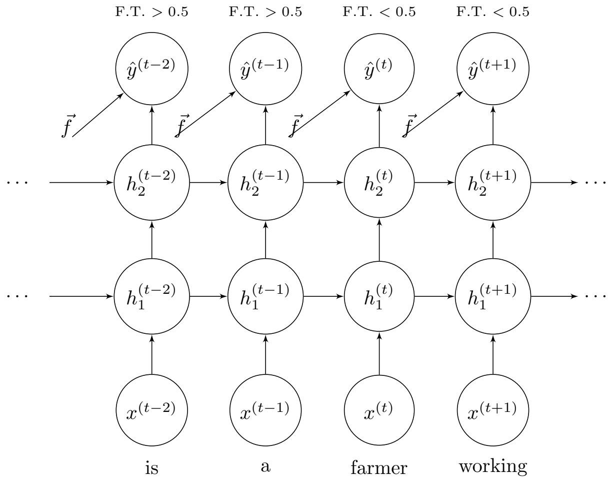

Figure 2: Network structure

We use the GloVe word vectors [5], trained on the Twitter corpus with dimension 25, as word embeddings. We have also tried word vectors with dimensions 50 and 100. However, it seems that higher dimensions do not yield significant gains in prediction accuracy.

## 5 Results

We compared several models, specifically,

- 2 layer RNN model with 256 and 144 units at each hidden layer, trained with number of steps 3;
- 2 layer LSTM model with 256 and 144 units at each hidden layer, trained with number of steps 3;
- 2 layer LSTM model with 256 and 144 units at each hidden layer, trained with number of steps 10;

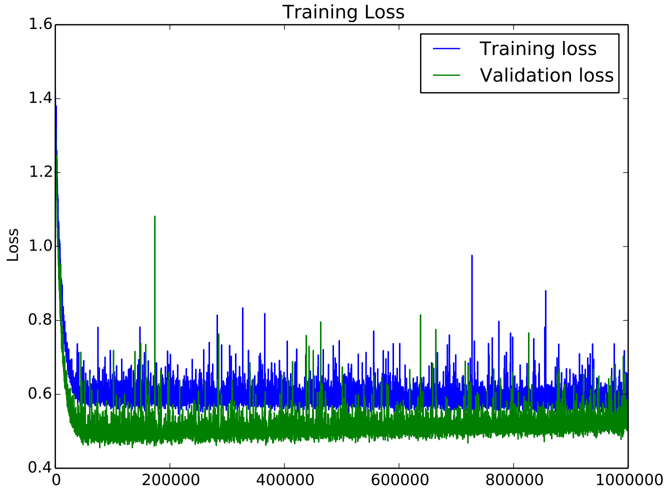

Figure 3: Loss plot during Training of LSTM

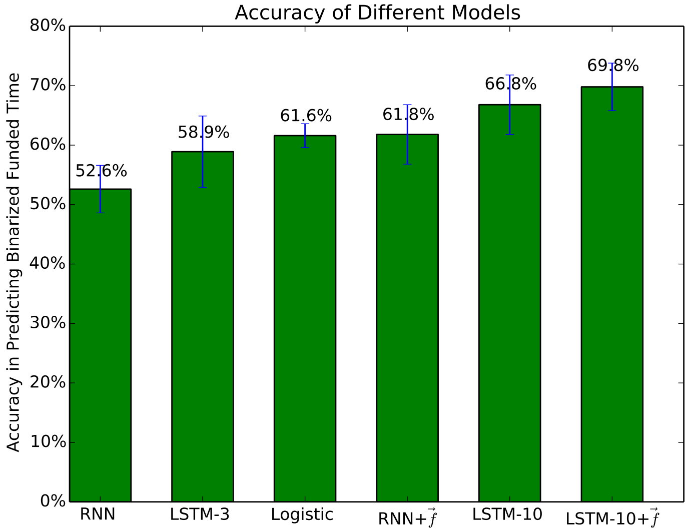

Figure 4: Result Comparison of Different Models

Moreover, for the aforementioned models, we incorporated other features as a feature vector $\vec{f}$, including loan amount, purpose of the loan and return period and so on, into the final projection layer, as shown in Figure 1. Models with and without this feature vector $\vec{f}$ at the final projection layer are also compared. The results are summarized in the chart below:

We notice that the recall and precision are nearly the same, as our classes are balanced. For simplicity, the performance metric is the overall prediction accuracy. From the above bar chart, we can see that the baseline logistic regression on other parameters achieves 61.8% accuracy. Using plain RNN trained with only the loan descriptions does not give much advantage as the accuracy is merely 52.6%. We suspect that the RNN does not capture the interaction between words, as the number of steps during training was only 3. The loan descriptions are typically in between 150 and 300 words, so it is very important to have a large window size to capture more interaction between words. Setting the number of steps longer than 5 will incur exploding gradient issue, which ruins the training process. Thus vanilla RNN does not fit well in our scope. Even when augmented with $\vec{f}$, the feature vector at the projection layer, RNN does not gain a significant edge than using logistic regression on $\vec{f}$ alone, as the accuracy was only 0.2% more, which is within the noise variance.

Using LSTM with longer training number of steps does give a significant improvement. When using LSTM cells for the middle 2 hidden layers, and trained with number of steps equals 10, the model is able to correctly predict the category of 66.8% of the loans, which is better than the benchmark logistic regression by 5%. Consider that the LSTM model used only the loan descriptions as the predictor, it is encouraging that it can already perform comparatively satisfying. Adding other features $\vec{f}$ at the projection layer further improved the accuracy by 3%, which already beats the benchmark logistic regression by 8%.

## 5.1 Visualization at Sentence Level

Our model outputs a softmax prediction at each word, which enables us to zoom in and look at the word-level predictions. Using the word level predictions, we can possibly infer which are the words that contribute to a fast loan funding. Below is an example of the output of an annotated loan description, with predicted labels in the parenthesis:

```
julio(0) t.(1) was(1) born(1) in(1) chibuto(1) district(0)
of(0) the(0) province(0) of(0) gaza(0) in(1) the(0) south(0)
of(0) mozambique(0) .(0) he(0) is(0) 43(0) years(0) old(0) ,
(0) married(0) and(0) father(0) of(0) six(0) children(1) aged(0)
one(0) through(0) 18.(1) of(0) these(0) ,(0) five(0) go(1) to(1)
school(1) .(1) he(1) finished(0) seventh(0) grade(0) and(0)
was(0) not(0) able(0) to(0) continue(0) his(0) studies(0) because(0)
of(0) financial(0) conditions(0) .(0) he(0) lives(1) in(0) his(0) own(1)
house(1) with(1) his(1) family(1) and(1) his(1) mother(1) .(1) he(1)
left(1) chibuto(1) in(1) 1983(1) for(1) maputo(1) and(1) the(1) home(1)
of(1) his(1) relatives(1) in(0) order(0) to(0) continue(0) with(0) his(0)
studies(1) .(1) after(1) his(1) marriage(1) ,(1) he(1) settled(1) down(1)
in(1) boane(1) ,(1) where(1) he(1) still(1) lives(1) today(1) .(1) he(1)
has(1) worked(1) as(1) a(1) civil(1) servant(1) in(1) the(1) department(1)
of(1) health(1) in(0) boane(0) for(1) more(1) than(1) 18(1) years(1) .(1)
he(1) has(1) a(1) salary(1) of(1) 2,900(1) mt(1) .(1) in(1) addition(1)
to(1) this(1) ,(1) he(1) provides(1) various(1) services(1) to(1) others(1)
for(1) which(1) he(1) earns(1) up(1)
```

### 5.1.1 Positive and Negative Words

We define a word to be positivity of a word to be

$$\text{Positivity}(w) = \frac{\text{Number of times } w \text{ being predicted positive}}{\text{Total number of occurrence of } w}$$

We picked the top 10 positive words and negative words. The words are chosen by their positivity defined above, and we chose words that occurred more than 100 times, in order to make our samples stable and avoid effects caused by randomness.

1. Positive:
   children (0.78), community (0.75), repayment (0.74), food (0.74), pass (0.68), home (0.67), fruit (0.66), married (0.66), morning (0.63), born (0.63)
2. Negative:
   school (0.33), old (0.34), business (0.34), village (0.34), years (0.36), loan (0.38), additional (0.39), costly (0.39), invest (0.39), expand (0.39)

It can be seen that the most positive words are related to families (home, children, married, born) and ). The negative words are more on the business side (business, loan, costly, invest, expand). It seems that although Kiva is a lending platform that is supposed to help people succeed in their businesses, the deterministic factors are still on the philanthropic side. This is a key insight on what type of loans are satisfied more quickly.

## 6 Conclusion and future work

In this project, we demonstrate that using recurrent neural networks with long-short term memory can beat the benchmark logistic regression model by a large margin, which suggests the loan description is indeed an important part in determining the loan funding time. Using appropriate LSTM model on loan description alone is able to beat logistic regression on all other features. We believe there is a greater role natural language processing can perform on online crowd funding platforms like Kiva, and we listed only a few of the possible follow-ups we will carry out for the rest of the project.

- Experiment with more possible approaches to predict funding time, including divide the labels into more bins and using square loss as the loss function;
- Use LSTM models to predict whether the loan will default;
- Provide feedback to the field partner on how to phrase a enticing loan description.

## References

[1] Ly P. and Mason G. (2012). Competition Between Microfinance NGOs: Evidence from Kiva In *World Development*, 2012, 40(3):643-655

[2] Liu et al. (2012). ”I Loan Because...”: Understanding Motivations for Pro-Social Lending. *WSDM'12*.

[3] Hochreiter S. and Schmidhuber J. (1997). Long short-term memory. *Neural Computation*, 1997, 9(8): 1735-1780.

[4] Mikolov et al. (2010) Recurrent neural network based language model. *Interspeech*, 2010, 2, 3

[5] Pennington J., Socher R. and Manning C. (2014) *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, p1532-1543