# Salary Classification and Applications Using Indeed Job Postings

**Sacha Rose Uritis**
Department of Statistics
srrobbin@uci.edu

**Sastha Kanagasabai Palaniappan**
Department of Computer Science
skpalani@uci.edu

**Yue Wang**
Department of Statistics
ywang47@uci.edu

## Abstract

Recent improvements in natural language processing technologies offer an opportunity to research and extract useful information from previously under-researched corpora of documents. We believe one such corpora are job postings (recruitment advertisements) in the 4 trillion-dollar information technology sector. We analyzed a subset of 14,867 documents (job posts) from the high-traffic job board, Indeed.com (Alexa traffic rank of 54 in US), looking for relationships between the documents and salary. From this preliminary review, we were able to determine that even with an imbalanced data set, we could predict salary up to a RSME score of 18.62 (in thousands of dollars) with an LSTM model and character encoding. Based on our study, we are optimistic about the utility of future studies and would like to offer the results of our study and recommendations for future research.

## 1 Introduction

It is estimated that 156 million people in the US participate in the labor market or receive some form of employment compensation [1], making the labor market in the US roughly 48 percent of the entire population.

And aside from its shear size and financial implications, labor market participation is something that nearly all adults will be involved with at some point in their life, and, therefore, contains a humanist reason for study. How many people's lives could be improved by a more efficient labor market? How much more efficient could our entire economy be? How could we improve our own job prospects post graduate school by understanding the labor market better?

Until very recently, research capabilities for analyzing aspects of the labor market such as how hard skills impact compensation, or what qualifications could build a career ladder would have been nearly impossible using manual statistical methods. However, modern natural language processing techniques offer a plausible method to answer questions like these and more.

So far, our project is unique to the world.

Our project aims to empower jobseekers. Using Indeeds job platform, we modeled the relationship between information on an online job post with the posted salary range. From this, jobseekers can learn about their worth in the job market based on their hard skills and location of interest. We hope jobseekers can use this information to improve their resumes and negotiate their salaries with employers.

### 1.1 Related Work

In this section, we will introduce some previous work that are related to our paper. In general, the relevant papers can be categorized into two fields: job title classification and job recommendation (or person-job match). The classification of job title based on job ads is a natural application of classification of natural language (NL) processing. Roa and Nino (2003) used Recursive Neural Networks (RNN) to perform classification of NL sentences and Kim (2014) experimented with CNN trained on top of word embeddings for sentence-level classification tasks. Based on these work, Xu et al. (2017) proposed a job posting corpus (JCTC) and used neural networks to get benchmarks for it. In terms of job recommendation projects, both conventional methods and neural networks have been explored. Zhang et al. (2015) leveraged collaborative filtering with user work background information to build a recommender system that recommends suitable jobs for its users. Zhu et al. (2018) proposed Person Job Fit Neural Network (PFJNN), which uses two CNNs

[1] US Bureau of Labor and Statistics

to encode resumes and jobs respectively and assess their match.

Moreover, Kim et al. (2015) proposed a CNN over characters, which outperforms word-level LSTM baselines with fewer parameters. This has shown the ability of character-level neural network to encode semantic information. We will use this technique, with a GRU instead, since hypothetically, this could provide a semantic word representation for technology terms.

## 2 Approach

In an effort to aid other researchers in reproducing our work we have uploaded all code and raw data into a Github Repository [2]. Pull requests are extremely welcome, so please let us know if you find any typos or have any suggested improvements to our approach.

### 2.1 Data Source Selection

To find a large enough dataset for exploration, we started by looking for career focused websites that offered enough traffic and postings that we could potentially harvest. Based on the top 10 career sites by traffic according to Alexa:

| Website | Alexa Ranking |
|---|---|
| Facebook | 2 |
| LinkedIn | 6 |
| Craigslist | 10 |
| Indeed.com | 54 |
| Stackoverflow | 61 |
| Monster.com | 147 |
| CareerBuilder.com | 168 |
| Glassdoor | 262 |
| SimplyHired | 489 |
| Dice | 836 |

After evaluating traffic, number of posts, authentication requirements, and ease of access to posts (via API/ scraping), we eventually decided that Indeed.com would provide the best opportunity to analyze job postings. Indeed is a high-traffic career website, they provide API access, their posts are all public, and they have no authentication requirements.

Originally, we hoped to extract job post data from Indeed's API, but eventually established that it was not a viable option, and elected to scrape job posts from the public web interface.

### 2.2 Legality of Scraping

After electing to scrape posts from Indeed for analysis we made sure to verify that we were legally allowed to do so. Since Indeed offers a public web interface, scraping indeed would fall under previous precedents set by several law cases: Nguyen v. Barnes & Nobel, Inc., Sandvig v. Sessions, hiQ Labs, Inc. v. LinkedIn Corporation, etc. As well, we believe our purposes would fall under fair use of copyright, as the analysis is for educational purposes.

### 2.3 Industry Selection

After we established Indeed.com as our data source, we next decided to narrow the scope of which job posts we would procure. The scope of our study warranted that we did not require additional data, nor would it make sense to utilize more resources to gather the additional data since our research is exploratory in nature.

To evaluate the various vertical markets against our chosen data source, we used the North American Industry Classification System (NAICS) and eventually identified the Information Technology, often called IT, sector as our primary focus of study.

IT offers several advantages over other vertical markets. First, we established in our initial research that ”hard” skills are easier to quantify and classify than ”soft” skills. For example, if a job post requires that an individual is certified to use an X-ray machine, we can quantify and parameterize that without much difficulty. However, if a job post requires that someone have a ”good attitude”, it is much more difficult to establish that a candidate qualifies, ergo difficult to use as a feature in a model. STEM fields, Medical, Finance, and IT all seemed to fit this profile.

Next, job titles in IT are generally related to the job duties/skills required and are fairly concrete. Some vertical markets such as Fast Moving Consumer Goods may refer to various positions as sales manager, director, or associate with the difference being explicit to the organization and not the industry at large. When we examined salary/compensation for such positions, we discovered that not only was salary varied (usually a base salary), but total compensation even for individuals with the same industry title varied widely. In IT, we found that most positions had a reasonable salary range associated with a title. As

[2] https://github.com/sacharose23/salary_classification_indeed_jobs

an example, a database administrator may differ by $30,000 between posts, but that the deviation was fairly normal between all positions.

We also found that industry dynamism (the pace of change) required aggressive learning curves and this was advantageous in contextualizing years of experience. Intuitively, this makes sense, of course. If a technology came out 5 years ago and an individual has 5 years experience they would have needed to be in the field before that time to be have the full duration of experience. Thus, years of experience with technologies can be considered an indirect indicator of proficiency level.

Lastly, we found in quick sample experiments that IT job posts lend themselves more easily to classification. We have theories for this, but the limited tests that we did were enough to help us narrow down our experiments.

### 2.4 Why Information Technology?

Information Technology, aside from making it easier to analyze, is a great industry to research. IT accounted for 5.2 percent of total GDP in 2015 and 2.7 percent of total employment. Furthermore, IT is estimated to have influenced total labor productivity growth by 6 percentages over the last 40 years. Contextually, this means that the labor market is incredibly important as it is both efficient and impactful for its size. [3]

Additionally, IT is particularly interesting for its growth trajectory. The number of jobs in the IT sector have been rapidly increasing within the last two decades, and the US Bureau of Labor and Statistics expect that software development and related jobs will increase about 30.7 percent in between 2016-2026. Out of the top fastest growing occupations, it is the second highest paying job (median 2017 : $101,790). They project that by 2026, there will be about 1 million new software developer jobs. So, it is no wonder why an IT career seems appealing to many individuals.

### 2.5 Data Acquisition and Feature Selection

Data collection was achieved by ”web scraping”, a processed, automated, method of generating requests, and then parsing the returned documents.

Originally, our stated plan was to use Indeed’s official API, but we later found out that access to official API was restricted to specific use cases. So, we resolved to build a web scraper that utilized a predetermined set of query parameters and Indeed’s search interface. Query parameters that were permutated:

| Parameter | Purpose |
|---|---|
| City | ensure corpus size |
| Search Radius | ensure corpus size |
| Job title | primary feature |
| Salary Range | dependent variable |

Ultimately, our dataset was made of 13 cities (San Francisco, Washington DC, New York, Seattle, Los Angeles, Philadelphia, etc.), chosen based on a study done by NerdWallet.com [4]. Each city was given a 100 mile radius as a query parameter to ensure that we had enough records matching the explicit job title for the search.

A list of 22 specific job titles was chosen for their availability and consistency across our data set. This list was seeded from Stack Overflow’s 2018 Developer survey [5], which polled over 100,000 developers worldwide, which contained questions about their job title and responsibilities. More details on exploratory analysis can be found in Appendix A.

We focused only on ’full-time’ jobs to reduce variation in salaries, and job titles were set to ’exact match’ in Indeed’s advanced search [6]to further reduce variation. Based on the salary distributions data in Indeed, we have chosen the lower and upper limits of salary ranges as $50k and $230k per annum (lowest salary from lowest paying IT jobs to highest salary from highest paying IT jobs from our dataset). Figure 1 shows the a fairly normal distribution of salary ranges in our dataset after collapsing all posts with salaries higher than $170k.

After data collection, the data curation process from raw data to structured data was implemented using key-value pairs (dictionary) for each document. The total dataset is a master list of dictionaries. For each data instance, the following primary fields were extracted: job title, salary range, city, hyperlink to job post source, and job description text. The hyperlink was used as the primary unique key to distinguish duplicates. After data de-duplication, we cleaned the data. Punctuations were removed using a custom-made to-be-removed punctuations list. Text was converted into lowercase. New line characters were removed

[3] Ian Hathaway

[4] Cities with the Most Tech Jobs

[5] Stack Overflow 2018 Developer survey results

[6] Indeed Advanced Search

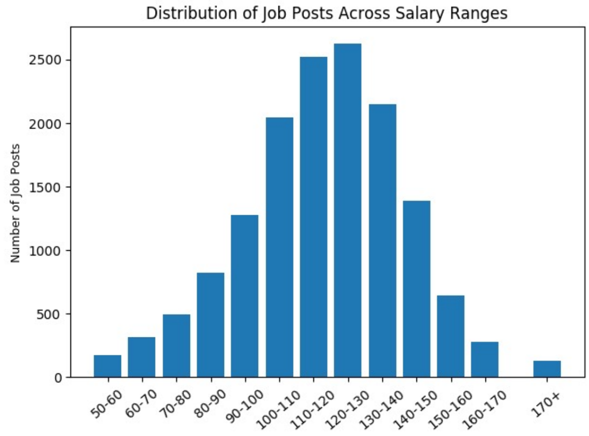


Figure 1: Distribution of Salary Ranges

for cleaner data to facilitate model training. After preprocessing, we had 14867 data points available for model training and evaluation.

## 3 Experiments

### 3.1 Baseline and Modified Logistic Models

We split our data into three sets with 70:20:10 ratio, respectively: training, testing, and validation. We used the validation set to evaluate our final model that had the best evaluation metric results for the testing set. Our baseline model for our multiclass classification is a logistic regression model. It will only have two basic features : city and job title. After we evaluated the baseline model, we modified it with other extracted features from the text in the job post description. We chose to extract keywords from a list of technologies (e.g. Python, Jupyter, R, Hadoop, Javascript, etc.) inspired by a community Github list [7], keywords from a list of education majors (e.g. computer science, mathematics, statistics, etc.) inspired by Stack Overflow 2018 Developer Survey, degree requirement (bachelors, masters, or doctorate), and years of experience required.

### 3.2 Deep Learning Model

The previous models are based on the flag values or counts of specific words, which are basically bag-of-word models. These kind of models ignore grammar and word order, which makes them suffer easily from insufficiency of information retrieval. What's more, it does not borrow any information from our prior knowledge of English words. To deal with these two problems, we improve our model in the following ways:

- To account for word order, we read in the whole description text of each job instead of just the multiplicity of key words. To feed the whole text into our model, we use a bidirectional LSTM layer to encode it.
- Transfer learning has been shown to be a great way of transferring knowledge from a related task that has already been learned. There are several versions of pre-trained semantic word embeddings for natural language tasks. Here we use GloVe.

These two improvements serve as a pretty good solution for general NLP tasks. However, it is important to notice that our corpus are different from the language we use everyday in terms that a large number of technology terms are used in job description corpus, such as JavaScript, SQL, HTML, to name a few. Most of them may not have a corresponding representation in any of the pre-trained word embeddings. But these tech terms usually play a crucial role in affecting salary, thus we don't want to simply treat all of them as unknown words. Here we use an embedding for every character in a word and feed them into a GRU layer to encode the word.

Two flow charts and a table of hyperparameters are included in Appendix C for a more detailed description of our model.

## 4 Evaluations

Our primary model validation techniques include normalized confusion matrices, precision, recall, and F1 scores, and root of mean squared error (RSME) on median of salary ranges. Confusion matrices ensure that for the given job description, the relevant salary range is predicted. Precision and Recall are useful evaluation methods for prediction when classes are imbalanced. The most optimal model will have a small RSME value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \qquad (1)$$

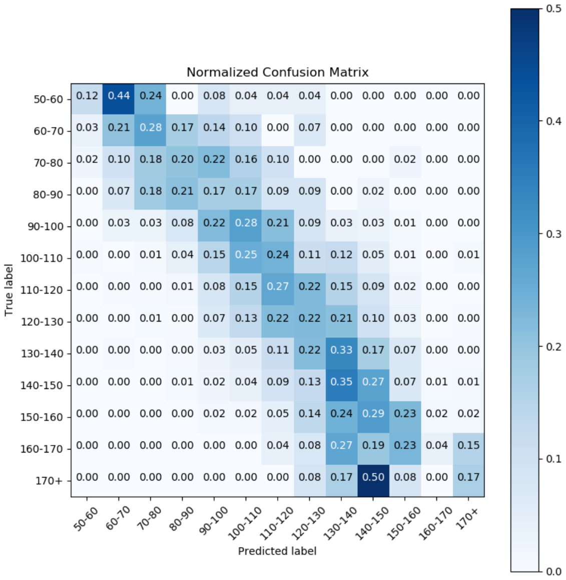
$$F_1 = \frac{2Precision \times Recall}{Precision + Recall} \qquad (2)$$

In the table above, the evaluation scores for Baseline and Modified Logistic models are based

[7] Github Awesome List

|  | Baseline Logistic | Modified Logistic | LSTM |
|---|---|---|---|
| Precision | 0.38 | 0.38 | 0.41 |
| Recall | 0.36 | 0.36 | 0.38 |
| F1 | 0.37 | 0.37 | 0.40 |
| RMSE | 19.94 | 19.57 | 18.62 |

on validation sets. Our final model, the LSTM model, shows evaluation metrics based on the hidden validation set. Our final model earned a RSME score of 18.62. Intuitively, this means that our final model predicts $18.62K away from the true median of salary range. Figure 2 shows a normalized confusion matrix of our final model, which is a better visual for analyzing accuracy across classes. For confusion matrices of the other two models, please refer to Appendix B.



| True label \ Predicted label | 50-60 | 60-70 | 70-80 | 80-90 | 90-100 | 100-110 | 110-120 | 120-130 | 130-140 | 140-150 | 150-160 | 160-170 | 170+ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50-60 | 0.12 | 0.44 | 0.24 | 0.00 | 0.08 | 0.04 | 0.04 | 0.04 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 60-70 | 0.03 | 0.21 | 0.28 | 0.17 | 0.14 | 0.10 | 0.00 | 0.07 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 70-80 | 0.02 | 0.10 | 0.18 | 0.20 | 0.22 | 0.16 | 0.10 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 | 0.00 |
| 80-90 | 0.00 | 0.07 | 0.18 | 0.21 | 0.17 | 0.17 | 0.09 | 0.09 | 0.00 | 0.02 | 0.00 | 0.00 | 0.00 |
| 90-100 | 0.00 | 0.03 | 0.03 | 0.08 | 0.22 | 0.28 | 0.21 | 0.09 | 0.03 | 0.03 | 0.01 | 0.00 | 0.00 |
| 100-110 | 0.00 | 0.00 | 0.01 | 0.04 | 0.15 | 0.25 | 0.24 | 0.11 | 0.12 | 0.05 | 0.01 | 0.00 | 0.01 |
| 110-120 | 0.00 | 0.00 | 0.00 | 0.01 | 0.08 | 0.15 | 0.27 | 0.22 | 0.15 | 0.09 | 0.02 | 0.00 | 0.00 |
| 120-130 | 0.00 | 0.00 | 0.01 | 0.00 | 0.07 | 0.13 | 0.22 | 0.22 | 0.21 | 0.10 | 0.03 | 0.00 | 0.00 |
| 130-140 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.05 | 0.11 | 0.22 | 0.33 | 0.17 | 0.07 | 0.00 | 0.00 |
| 140-150 | 0.00 | 0.00 | 0.00 | 0.01 | 0.02 | 0.04 | 0.09 | 0.13 | 0.35 | 0.27 | 0.07 | 0.01 | 0.01 |
| 150-160 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.02 | 0.05 | 0.14 | 0.24 | 0.29 | 0.23 | 0.02 | 0.02 |
| 160-170 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 | 0.08 | 0.27 | 0.19 | 0.23 | 0.04 | 0.15 |
| 170+ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.08 | 0.17 | 0.50 | 0.08 | 0.00 | 0.17 |

Figure 2: Normalized Confusion Matrix for Deep Learning LSTM Model

## 5 Conclusions and Future Work

Since our classes were in salary range form as opposed to non-exact salary numbers, we believe our error range is adequately successful. Our final model accurately predicts salary plus or minus two classes from the truth. We believe our model can be improved with more data from other job posting platforms, exact salary specification for the ultimately acquired job position, and a standardization of certain label features.

For potential future work, we recommend techniques in placating the effects of imbalanced data. Imbalancement across classes is a common challenge in classification. Imbalancement causes heavy biases toward highly represented classes. There are several ways to remedy the effects: resampling techniques, which include undersampling and oversampling, like Synthetic Minority Over-sampling Technique (SMOTE), and Then, adding biases toward the minority classes using class weights.

## References

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models. arXiv:1508.06615. Version 4.

Sergio Roa and Fernando Nino. 2003. Classification of natural language sentences using neural networks.

Haoyu Xu, Chongyang Gu, Han Zhou, Sengpan Kou, and Junjie Zhang. 2017. Jctc: A large job posting corpus for text classification. arXiv:1705.06123. Version 1.

Chen Zhu, Hengshu Zhu, Xiong Hui, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. 2018. Person-job fit: Adapting the right talent for the right job with joint representation learning. arXiv:1810.04040. Version 1.

## A Exploratory Analysis

| Technology | No of Job Posts |
|---|---|
| Python | 5088 |
| AWS | 3316 |
| Java | 3030 |
| Linux | 2751 |
| Database | 2615 |
| Javascript | 2381 |
| R | 2193 |
| Go | 1927 |
| Spark | 1792 |
| Git | 1666 |

Figure 3: Top 10 Technologies in Scraped Dataset

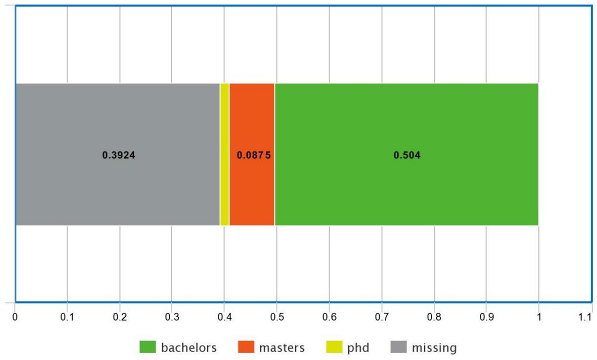


Figure 4: Proportion of Education Degrees from Dataset

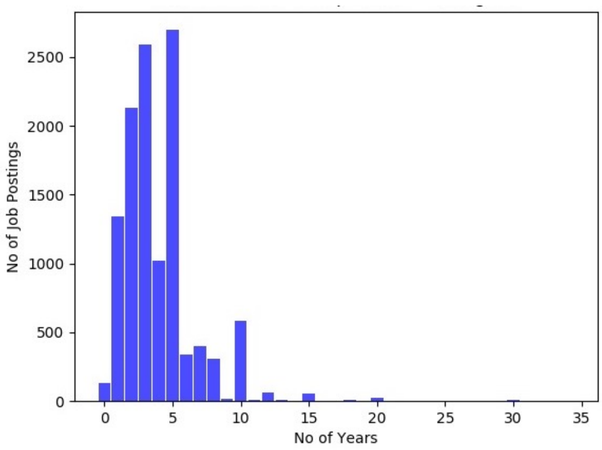


Figure 5: Minimum Years of Experience from Dataset

## B Confusion Matrices of Logistic Regression

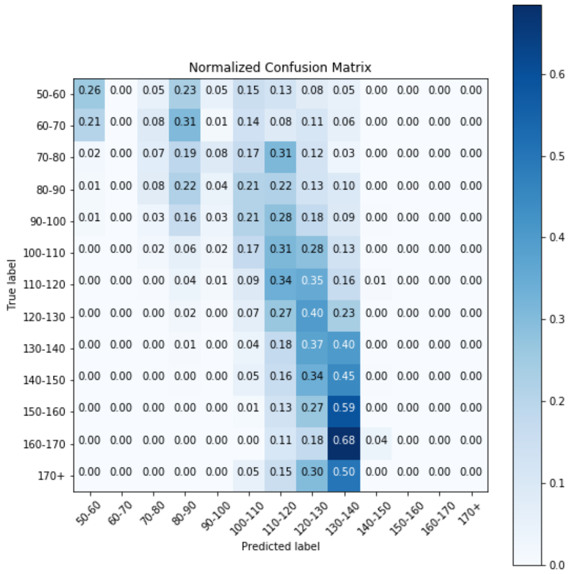


Figure 6: Normalized Confusion Matrix for Baseline Logistic Regression Model

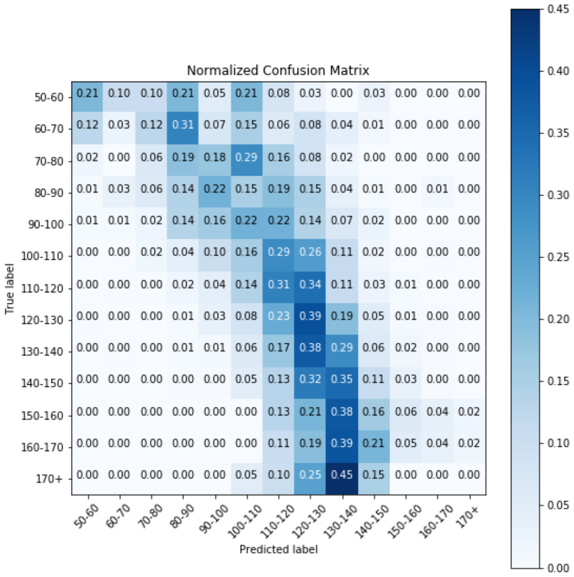


Figure 7: Normalized Confusion Matrix for Modified Logistic Regression Model

## C Neural Network Setup

| | Embedder | | Encoder |
|---|---|---|---|
| | GloVe | GRU | LSTM |
| input size | - | 10 | 75 |
| hidden size | 50 | 25 | 40 |
| bidirectional | - | No | Yes |
| dropout | - | 0 | 0.25 |

Table 1: NN hyperparameters

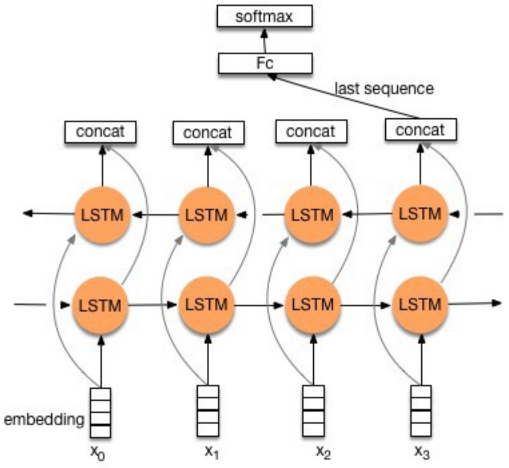


Figure 8: NN model with bidirectional LSTM

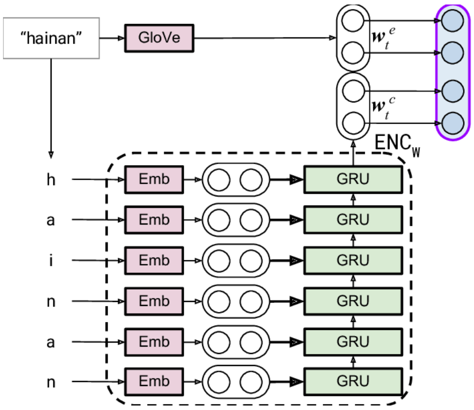


Figure 9: Embedding layer with character-level GRU encoder