Short Text Classification Using Semantic Random Forest

Conference Paper · September 2014

DOI: 10.1007/978-3-319-10160-6_26

CITATIONS
39

READS
1,303

5 authors, including:

Christel Dartigues-Pallez
University of Nice Sophia Antipolis
22 PUBLICATIONS **168** CITATIONS
SEE PROFILE

Celia da Costa Pereira
University of Nice Sophia Antipolis
117 PUBLICATIONS **821** CITATIONS
SEE PROFILE

Frederic Precioso
University of Nice Sophia Antipolis
153 PUBLICATIONS **2,163** CITATIONS
SEE PROFILE

Short Text Classification Using Semantic Random Forest

Ameni Bouaziz[1], Christel Dartigues-Pallez[1], Célia da Costa Pereira[1], Frédéric Precioso[1], and Patrick Lloret[2]

[1] Laboratoire I3S (CNRS UMR-7271), Université Nice Sophia Antipolis
bouaziz@i3s.unice.fr,
{christel.dartigues-pallez,celia.pereira}@unice.fr,precioso@polytech.unice.fr
[2] Semantic Group Company, Paris
plloret@succeed-together.eu

Abstract. Using traditional Random Forests in short text classification revealed a performance degradation compared to using them for standard texts. Shortness, sparseness and lack of contextual information in short texts are the reasons of this degradation. Existing solutions to overcome these issues are mainly based on data enrichment. However, data enrichment can also introduce noise. We propose a new approach that combines data enrichment with the introduction of semantics in Random Forests. Each short text is enriched with data semantically similar to its words. These data come from an external source of knowledge distributed into topics thanks to the Latent Dirichlet Allocation model. Learning process in Random Forests is adapted to consider semantic relations between words while building the trees. Tests performed on search-snippets using the new method showed significant improvements in the classification. The accuracy has increased by 34% compared to traditional Random Forests and by 20% compared to MaxEnt.

Keywords: Short text classification, Random Forest, Latent Dirichlet Allocation, Semantics.

1 Introduction

In the past few years, an expansion of interactive websites usage has been seen allowing users to contribute to their content. The users contributions are generally short texts (comments in social networks, feedbacks in e-commerce websites, etc.). This has led to huge amounts of stored short texts with an increasing need to classify them and to extract the knowledge they contain.
Traditional text classification process usually implies 3 steps:

- *Pre-processing among which feature extraction from texts*: a set of words are carefully chosen from the texts in order to be used in the document representation. The bag-of-words model is the most common process for this purpose. In such a model, the occurrence (or frequency) of each word in the document is used for representing the documents.

- *Learning*: the set of document representations obtained after the pre-processing phase is used for training a classifier (or classification model).
- *Classification*: in this last step the classification model is used on new texts to determine which class they belong to.

As pointed out by [1] and others, the performance of using traditional text classification methods, (like random forest for example), for short text classification is limited, because of the word sparseness, the lack of context information and informal sentence expressiveness. A common method to overcome these problems is to enrich the original texts with additional information. However, data enrichment can also produce noise and it is then important to take the real benefit of these new data in the classification process into consideration before using them.

Some authors [1–4] proposed new methods which are essentially based on the enrichment of short texts , on "intelligent" reduction of the number of irrelevant document features and/or on the combination of them. However, none of those methods proposed to combine a semantic enrichment both for each word in the text and for the text as a whole entity. Furthermore, to the best of our knowledge there is no work in the literature exploring the semantic behind the short texts through out the construction of the trees in a random forest.

We propose a new Random Forest method impacting on the first two previously mentioned classification steps as follows:

- At the pre-processing step, we use the Latent Dirichlet Allocation (LDA)[11] model to derive topics from existing texts. The short texts are then enriched, in a two-level process, in order to increase the number of features. More precisely, first, the text is enriched with the topics that are more similar to each word in the text. Besides, in order to also take into consideration both the overall context information and the informal expressiveness of the short text, we further enrich the text with the words from the four topics which are more relevant with respect to the whole text.
- At the learning step, the construction of the trees is based on a random selection of the features obtained in the previous step. The information gain concerning these features determines the feature to be chosen at each node. We adapt the learning step in order to also exploit the semantic relations between the features.

The above two-level enrichment provides a first improvement in the quality of the classification model. Exploiting semantics when constructing the trees provides a further improvement. These are the two original contributions of our work.

The rest of the paper is organized as follows. Section 2 presents related works, Section 3 introduces our classification method. Experimental results are presented in Section 4. Finally, Section 5 concludes the paper and presents some potential ideas for future work.

2 Related Works

Random Forest introduced by Breiman in [5] are one of the most known and effective machine learning algorithms in data mining and particularly in text classification. They base their classification on a majority vote of decision tree classifiers. Many implementations were proposed in the literature depending on the nature of data to classify.

Forest Random Combination [5] is an extension of the traditional implementation that creates new features by combining linearly existing ones, this helps in increasing the size of the features space in case of small datasets. Geurts et al. proposed in [6] the Extremely Randomized Trees where they modify the feature selection process by making it totally random instead of based on a partitioning criterion. Geurts et al. increase diversity in trees but make them bigger and more complex. To solve classification problems for unbalanced datasets, Chen et al. in [7] proposed two solutions: Balanced Random Forest, based on sampling to balance data, and Weighted Random Forest which assigns weights to features according to their representativeness. This weighting is taken into account at feature selection and at majority votes.

All these algorithms gave good performances when applied on data they were designed to classify. However, when it comes to short text classification, they are less efficient, indeed short texts have many characteristics that make their classification a challenge. The most obvious characteristic is the shortness of the texts [8] because they are composed of few words up to few sentences at maximum (tweets, for instance, do not exceed 140 characters). As a consequence those texts do not provide enough data to measure similarities between them. For example, there is no way to know that two short texts, one composed of synonyms of the other, share the same context without considering semantic relations between the words which compound them.
Short texts are characterized by their data sparseness [8]. As input of learning algorithms a text may be represented by a vector containing the weights of its words. Because of that sparseness, vectors representing short texts are nearly empty. This lack of information implies a poor coverage of the classes representing the dataset, thus complicating the correct classification of new data.

Most of the works on short text classification propose to use semantics to overcome these challenges by either reduction or enrichment of features. Both of them use an external source of knowledge to provide a semantic network that builds relations between features. The semantic network can be an ontology (WordNet or Wikipedia) [9, 10] or it can use topic model techniques (LDA, Latent Semantic Indexing [12], ...) on large scale data collections to semantically group them into topics.

Yang et al. [1] come with a reduction approach based on topic model, they combine semantic and lexical aspects of short text words. With this approach the feature space dimension is no longer equal to the whole number of words but is reduced to the number of topics. A mapping based on semantic and lexical features allows to transform the vector representing a text in the word space

to a smaller representation in the topic space. This method achieved efficient classification, however it seems to be only efficient on equally distributed data over the different classes.

In [8] Phan et al. introduce a new method of short texts enrichment, they apply LDA to generate topics, then they integrate these topics in the short texts. This method combined with the Maximum Entropy learning algorithm [13] gave a high classification accuracy even when tested on small size datasets. However as pointed out by Chen et al. in [3] the Phan's approach [8] may have limits owing to the fact that generated topics are considered of a single level. To tackle this problem, Chen et al. propose an algorithm that selects the best subset of all generated topics. That subset contains topics which help more discriminating short text classes while being separated enough from each other. They therefore ensure that short text enrichment is done only based on relevant topics and that no noise is introduced by adding less important topics.

All works above improve short text classification but we noticed that they focused exclusively on the texts pre-processing phase, no study has been carried out to adapt the various learning algorithms to use semantics during the classification phase, in particular with random forests. In some works interested in Random Forest prediction tasks like [14] and [15], some attempts have been made to benefit from semantic relations between features during the learning process.

Caragea et al. in [14] organize features in a hierarchical ontology and combine it with four learning algorithms to predict additional friendship links in a social network. Best results were obtained with Random Forests. In [15] Chen and Zhang propose a new Random Forest implementation named module Guided Random Forests in order to predict biological characteristics. They create a correlated network of features and group them into modules which are used to guide the feature selection in all the nodes of the trees.

Although these works are not really linked to text classification , their results on prediction tasks encourage to implement a new Random Forest method that takes benefit from semantic links between words to improve short text classification.

3 Description of the Proposed Method

In this section we will describe in detail: (i) how the short texts are enriched, (ii) how the document's features are selected, and (iii) how the semantic of the words are considered in the Semantic Random Forests.

3.1 Short Text Enrichment

To overcome the short texts classification issues mentioned above, like in [8], we propose to enrich short texts with words semantically related to their words. Thus we transform the vector representing a short text in a larger one containing more information.

The proposed enrichment method relies on the semantic relations between the words in the short texts, we use an external semantic network as a source for these relations. The network must satisfy two conditions, first it has to be specific to our dataset and second it has to cover all our data domains. To build this semantic network we apply the Latent Dirichlet Allocation (LDA)[11] on an external source of data. LDA is a topic model method that allows to discover underlying structure of a big set of data. It is a generative probabilistic method that uses Dirichlet distribution to identify hidden topics of a dataset. Then, based on similarity calculation it associates each word to one or many of the generated topics. Each word has a weight representing its importance in a topic. Thanks to LDA we obtain a semantic network where nodes are topics and words, and edges are links between topics and word. The edge's weight is the corresponding word weight in the topic. The obtained network is used in enriching short texts following this algorithm:

Enrichment Algorithm

```
program EnrichTexts (Output: set of  enriched texts)
   Input: set of short texts, semantic network
   for shortText in short texts:
      // words enrichment Procedure
      topicsForAllTextWords = []
      for each word in shortText:
         topicWeight = [] //map of topics and the
         //weight of the current word in them
         for each topic in topics:
            if word in topic:
               topicWeight[topic] = wordWeight
            else:
               topicWeight[topic] = 0
         bestTopicForWord = max(topicWeight)
         topicsForAllTextWords += bestTopicForWord

      // whole short text enrichment procedure
      similarities = []
      bestTopics=[]
      for topic in topics:
         similarity = computeSimilarity(shortText,topic)
         similarities.add(similarity)
       bestTopics = find_N_BestTopics(similarities)

      // Adding found topics to current short text
      for topic in topicsForAllTextWords:
         shortText.add(topic)
      for topic in bestTopics:
         shortText.add(topic)
end.
```

To enrich a short text, our algorithm considers the short text into 2 different ways:

- a text as a set of words taken separately: for each word, the algorithm looks for the nearest topic to the word, which means the topic in which this word has the biggest weight. Then we add to this word in the short text all the words of the found topic. This process transforms short texts in a set of general contexts. It allows to build a generic model that is able to classify any text related to the domains of the initial short texts,
- a text as one entity: the goal here is to give more importance to the general meaning of the text. The algorithm computes the similarities between all the topics and the text. The similarity is defined as the number of common words between a text and a topic. Then the algorithm chooses the N nearest topics, which means the N topics with the highest similarities. Finally, the algorithm adds all the words of the chosen topics to the text.

To illustrate the enrichment algorithm, let us imagine we want to enrich the following short text (obtained after lexical pre-processing):

national football teams world

The enrichment will rely on a set of four topics defined as follows (the numbers are the weights of the words in the topics):

- Topic 1: sport (0.6), football (0.2), teams (0.15), goal (0.05).
- Topic 2: music (0.4), instrument (0.3), songs (0.2), piano (0.1).
- Topic 3: volleyball (0.5), teams (0.3), win (0.12), basketball (0.08).
- Topic 4: web (0.45), site (0.3), programs (0.2), computer (0.05)

Words *national* and *world* do not belong to any topic, they are not enriched, however the word *football* is enriched with topic 1 as this is the only topic it belongs to. The word *teams* belongs to topics 1 and 3, but only topic 3 is used in enrichment as the weight of *teams* in topic 3 is bigger than its weight in topic 1. After enrichment at word level the short text becomes:

national football *sport football teams goal* ***teams*** *volleyball teams win basketball* ***world***

For the whole text enrichment process, if we consider adding the two nearest topics to the texts, topics 1 and 3 are used as they contain the biggest number of common words with the text. At the end, the enriched text is:

national football *sport football* ***teams*** *goal teams volleyball teams win basketball* ***world*** *sport football teams goal volleyball teams win basketball*

3.2 Semantic Random Forest

After having enriched short texts, we focus now on the Random Forest learning process. Random Forest are sets of decision trees whose nodes are built from a set of features obtained at pre-processing step. In our case features are all the words of the enriched texts. We propose a new implementation named Semantic Random Forest that reduces the random feature selection in favor of a semantic driven feature selection. Indeed, in traditional Random Forest, all the features of the corpus are used in building the trees, for each node the algorithm selects randomly K features, then it calculates information gain of each feature (using Gini criterion or Entropy). The feature that provides the best gain is chosen for the node. In Semantic Random Forest this process is slightly modified to become semantically driven: our goal is to have trees whose nodes are semantically linked. We need in a first step to organize the whole set of features in a way where those that are semantically linked are grouped together. We use again LDA to group features into topics. Again, each feature is assigned to a weight representing its importance in a topic. In a second step we modify the tree building process as follows: instead of using all the features to construct a tree, the algorithm makes a first random selection of a small set of features, then for each of these chosen features it looks for the topic where this feature has the maximum weight, then it adds from this topic all features that have weights bigger than a given threshold. Finally the tree is built based on this new set of features by applying the same random feature selection method.

This method allowed us to obtain trees composed of nodes belonging to the same topics, so semantically linked, nevertheless, the initial small feature set selection ensures that we keep the randomness of the forest and avoid then correlation between trees. It also ensures that features not belonging to any topic are not discarded.

Semantic Random Forest

```
program SemanticRandomForest (Output:Semantic Trees)
   Input: set of features extracted from enriched texts
          L, //number of features to be chosen randomly
          K, //total number of features
          threshold //minimum weight of considered words
   // group all features semantically into topics
   Topics = apply LDA on the features set
   For each tree in trees:
      select L features < K
      // build a new set of semantically linked
      //features and store it totalList
      totalList= []
      For each feature in features:
         // look for features from the nearest topic to
         //the current feature
         featuresToAdd = []
         featuresToAdd  =  SFS (feature, topics, threshold)
```

```
          totalList += featuresToAdd
      Construct tree using totalList
      //using standard tree building algorithm
end.
```

Semantic Features Selection

```
program SFS (Output:featuresList)
   Input: feature, topics, threshold
   nearestTopic = FindNearestTopic (feature,topics)
   //the topic in which the feature has the biggest weight
   for each feature in nearestTopic:
      if feature.weight >= threshold
         featuresList += feature
end.
```

4 Experiments and Results

4.1 Experimental Dataset and evaluation criterion

To validate our new method we tested it on the "search snippets" dataset. This dataset was collected by Phan [8] and is composed of:

– *short texts corpus*: built from top 20 to 30 responses given by Google search engine for different queries. Each response is composed of an URL, a title and a short description. Each short text is labeled by a class according to the submitted query. The whole corpus contains 8 categories and is divided into training data and test data as shown in table 1.

Table 1. A summary of the distribution of the short texts in the corpus.

Categories	**Train**	**Test**
Business	1200	300
Computer	1200	300
Culture Art Entertaiment	1880	330
education science	2360	300
engneering	220	150
Health	880	300
Politics Society	1200	300
Sport	1120	300

– *Universal dataset*: this dataset is composed of a set of documents collected from Wikipedia as a response to queries containing some specific keywords. The application of LDA on this document set generated 200 topics with

200 words each. We used these topics as a semantic network for short text enrichment.

To evaluate our algorithm we used the accuracy, defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:
TP: Number of true positive; TN: Number of true negative
FP: Number of false positive; FN: Number of false negative

4.2 Results and Interpretations

Before applying Semantic Random Forest on search snippets dataset, we tested in a first step the enrichment of texts taken as whole entities combined with traditional Random Forest. The goal is both to evaluate this enrichment's contribution in improving the accuracy and also to find the best number of topics to be used for enriching the short texts. We run tests using a traditional Random Forest implementation provided in the scikit learn library [16], the tests were run first on the search snippets texts without enrichment, then enriched with respectively 1, 2, 3, 4 and 5 topics. All this was repeated using 10 then 20 up to 100 trees. Table 2 and figure 1 summarize the obtained results.

Table 2. Classification accuracy depending on trees number with enrichment of 1 to 5 topics.

	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
No Enrichment	0.57	0.592	0.598	0.589	0.589	0.592	0.594	0.595	0.6	0.59
1 topic	0.64	0.643	0.669	0.68	0.679	0.681	0.679	0.684	0.678	0.68
2 topics	0.662	0.666	0.667	0.668	0.674	0.675	0.675	0.675	0.673	0.676
3 topics	0.66	0.667	0.671	0.673	0.675	0.672	0.67	0.672	0.672	0.67
4 topics	0.674	0.696	0.697	0.7	0.704	0.706	0.705	0.707	0.703	0.707
5 topics	0.675	0.693	0.694	0.7	0.702	0.705	0.705	0.705	0.703	0.702

These results show that enriched texts classification is better than short texts classification whatever the number of trees considered. Indeed, the accuracy increased from 0.59 to 0.707 when using 100 trees and an enrichment of 4 topics. As we can see in the graph above, the experience shows also that the best number of topics to add to each short text is 4. Indeed, starting from the fifth topic, texts and topic become semantically far from each other, adding topics is then a noise introduction rather than an enrichment. We evaluate now the added value of the full enrichment (text as an entity and as a set of words) and the semantic random forest. We enrich the text by our two enrichment processes: adding the 4 nearest topics to whole text and nearest topic to each word. We apply both traditional and semantic random forest on these enriched texts. These tests are

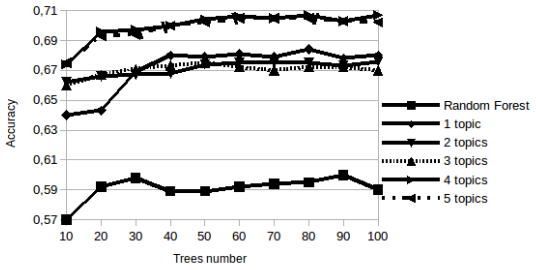

Fig. 1. Variation of short texts classification accuracy depending on trees number for short texts enriched from 1 to 5 topics.

repeated 10 times varying the tree number each time. The obtained results are shown in the table 3 and figure 2.

Table 3. Classification accuracy depending on trees number with Random Forest, Random Forest and enrichment and with Semantic Random Forest.

	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Random Forest	0.57	0.592	0.598	0.589	0.589	0.592	0.594	0.595	0.6	0.59
Enrichment only	0.728	0.745	0.753	0.755	0.758	0.758	0.76	0.761	0.76	0.766
Enrichment + SRF	0.73	0.761	0.771	0.776	0.7763	0.78	0.784	0.784	0.786	0.789

The results in table 3 show the classification improvement obtained thanks to our two enrichment methods. By combining whole text enrichment and word by word enrichment the accuracy increased up to 0.766 for a test with 100 trees, this is higher than the accuracy of 0.707 obtained in our first experiment where only whole text enrichment was done. After this second test, compared to classification without enrichment, we achieved already an accuracy increase from 0.59 to 0.766 which represents a 30% improvement.
Applying the Semantic Random Forest instead of the traditional Random Forest on the same enriched texts, we obtained our best classification results, the accuracy reached the value of 0.789 for 100 trees. SRF allowed an additional improvement of 3% compared to the classification of enriched texts using traditional Random Forests. Our method combining text enrichment and SRF allowed a total increase of 34% compared to traditional short text classification. Results show also that almost the same improvement rates were obtained regardless of the forest trees number which confirms the efficiency of our method.
The classification of the search snippets dataset was also tested on the Maximum Entropy method [13]. The accuracy obtained with MaxEnt was 0,657 which is better than traditional Random Forest but lower than the accuracy obtained by applying our method whose improvement reaches 20%.

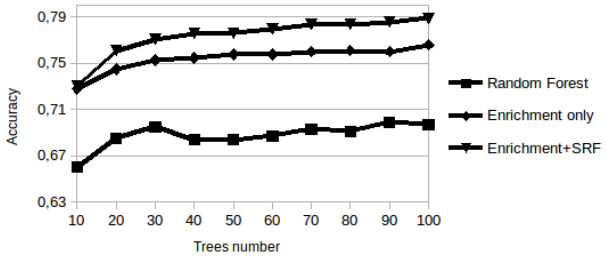

Fig. 2. Variation of short texts classification accuracy depending on trees number for short texts with traditional Random Forest, enriched texts with traditional Random Forest and Enriched texts with enriched Random Forest.

In addition to the classification improvement, our method allowed a significant reduction in the trees size. Table 4 shows the evolution of the trees size concerning the 10 trees test. We can notice that the average number of the nodes of Semantic Random Forest trees (2826) is less than the half of those of Random Forest trees(6272).

Table 4. Number of tree nodes for Random Forest and Semantic Random Forest.

Tree	1	2	3	4	5	6	7	8	9	10	average
Nodes in RF	6355	6451	6473	6119	5935	6143	6193	6339	6061	6657	6272
Nodes in SRF	2957	2883	2657	2653	2879	2851	2873	2717	3037	2753	2826

5 Conclusion and Future Work

In this work, we proposed a new two-step short text classification approach. The first step is dedicated to text enrichment. The enrichment algorithm relies on Latent Dirichlet Allocation for generating, from an external source of data, topics which are further added to the short text. In our method, topics are considered at word-level, for their similarities with the words in the text, and at text-level for their relevance with the whole text content. The second step is our new Random Forest algorithm that we call Semantic Random Forest (SRF). The SRF learning process is different from the traditional Random Forest in that it considers the semantic relation between features to select the features involved in decision tree construction. We applied our method to the search-snippets dataset and obtained results showing a classification improvement which reaches 34% compared to traditional Random Forest and 20% compared to Maximum Entropy method.

We provide a further improvement by exploiting semantics when constructing the trees. We think that by introducing semantic relations at node-level for

feature selection could lead to new improvements. Our semantic-based approach is also fully compliant with the Weighted Random Forest approach, this could be an extension of our work for unbalanced datasets classification. It would be also interesting to apply our method on different datasets. This will allow us to explain how to generally define the parameters of our algorithms and study their complexities.

Acknowledgments. This work is co-funded by Région Provence Alpes Côte d'Azur (PACA) and Semantic Grouping Company (SGC).

References

1. Yang, L., Li, C., Ding, Q., Li, L.: Combining Lexical and Semantic Features for Short Text Classification. In 17 th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES (2013)
2. Amaratunga, D., Cabrera, J., Lee, Y.S.: Enriched Random Forests. Bioinformatics. 24,18,2010–2014 (2008)
3. Chen, M., Jin, X., Shen, D.: Short Text Classification Improved by Learning Multi-Granularity Topics. 22nd International Joint Conference on Artificial Intelligence (2011)
4. Song, Y., Wang, H., Wang, Z., Li, H., Chen, W.:Short Text Conceptualization using a Probabilistic Knowledge base. 22nd International Joint Conference on Artificial Intelligence, pp 2330–2336 (2011)
5. Breiman, L.: Random Forests, Machine Learning, 45, 5–32 (2001)
6. Guerts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach Learn. 63,3–42 (2006)
7. Chen, C., Liaw, A., Breiman, L.: Using Random Forest to Learn Imbalanced Data (2004)
8. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. In www 2008 Data Mining-Learning, Beijing, China (2008)
9. Hu, X., Zhang, X., Caimei L., Park E.K., Zhou, X.: Exploiting Wikipedia as External Knowledge for Document Clustering. In:KDD'09, Paris, France (2009)
10. Hu, X., Sun, N., Zhang, C., Tat-Seng, C.: Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge. In: CIKM'09, pp. 2–6, Hong Kong, China (2009)
11. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. Journal of Machine Learning Research. 993–1022 (2003)
12. Dumais, S.T.: Latent Semantic Indexing. In TExt REtrieval Conference, pp 219–30 (1995)
13. Berger, A., Pietra, A., Pietra, J.: A maximum Entropy Aproach to Natural Language Processing. Computational Linguistics. 22(1), 39–71 (1996)
14. Caragea, D.,Bahirwani, V., Aljandal, W., Hsu, W.: Ontology-Based Link Prediction in the LiveJournal Social Network. 8th Symposium on Abstraction, Reformulation and Approximation (2009)
15. Chen, Z., Zhang, W.: Integrative Analysis Using Module-Guided Random Forests Reveals Correlated Genetic Factors Related to Mouse Weight. Plos Computational Biology. 9, e1002956 (2013)
16. Scikit-Learn Machine Learning in Python, http://scikit-learn.org