Predict the emergence - Application to competencies in job offers

Yacine Abboud, Anne Boyer, Armelle Brun

Predict the emergence
Application to competencies in job offers

Yacine Abboud*, Anne Boyer*, Armelle Brun*
*Université de Lorraine, LORIA UMR 7503
Nancy, FRANCE
Email: yacine.abboud@gmail.com, anne.boyer@loria.fr, armelle.brun@loria.fr

***Abstract*—Predicting the emergence of an event enables to anticipate and make decisions upstream. For instance, in the employment sector, it becomes necessary to anticipate the emergence of competencies requirements to help job seekers, education and training organization to better match the needs of the job market. Several approaches address the competencies mining with ontologies, we adopt a different point of view by using pattern mining. We propose a new methodology to predict emerging patterns and apply it to competencies with a dataset of job offers collected on the Web. Our model allows to identify potential emerging pattern over time and thus enables to take decisions accordingly.**

***Keywords*-Emergence; Pattern mining; Competencies; Web mining;**

I. Introduction

In many sectors, the ability to anticipate future trends is crucial to make decisions in accordance with the evolution of the environment. Regarding the employment, it becomes necessary to anticipate any evolution about jobs to allow job seekers to adapt by training. Indeed, jobs are constantly evolving and the competencies required for those jobs are evolving too. That is why the identification of any new trend regarding required competencies in the job market will enable job seekers, education and training organizations to better match the market's needs. If those organizations are able to detect emerging competencies, whether existing or new, they will be able to train and provide training on those competencies. However, predicting emerging competencies is a real challenge both for the detection of competencies in unstructured data sources and for the prediction of emergence. As a competency is merely a pattern of words in a given context, we can view a competency as a pattern [Agrawal and Srikant, 1995] and use pattern mining methods. A large part of our work was to adapt pattern mining to competencies. A specific pattern type appeared to be more relevant for competencies: an n-gram [Fürnkranz, 1998], which is a contiguous pattern.

Emerging pattern has been definied by [Dong and Li, 1999] to determine if patterns are emerging between two datasets. This definition lacks the notion of time and does not include more than two datasets. To the best of our knowledge, the prediction of emerging pattern over time is still unstudied, unlike the prediction of the emergence. [Thorleuchter et al., 2014] detects future emerging sequences of words in response to a scientific hypothesis on the Web. [Minh-Hoang Le, 2005] detects emerging trends in a set of scientific articles. Those methods are based on word frequency evolution over time and linear regression. In contrast to related works [Thorleuchter et al., 2014][Minh-Hoang Le, 2005], our methodology is based on pattern mining [Agrawal and Srikant, 1994] and linear regression to predict the emergence of patterns over time.

Our contribution is the conception of a model to predict emerging competencies using linear regression and n-gram mining. We assess our model on a French job offers website to predict emerging competencies. Our goal was to check the ability to predict emergence and extract competencies. This model shows a competency mining with a new point of view, through activities, and a new way to predict emerging pattern.

In this paper, we will describe the main contributions regarding the emergence and competencies. Before showing how our model can provide a solution to predict emerging competencies and illustrate our methodology with a case study. Finally, we will conclude with an evaluation and a presentation of future leads.

II. Background

A. Competencies

Organizations have to adjust faster and faster to stay competitive. Employees must be able to adapt and make evolve their competencies through time. That is why competencies are the core of organizations ability to adjust to continuous developments.

According to [Wittorski, 1998], "a competency corresponds to the mobilization in the action of a number of combined knowledge depending on the specific context built by the author of the situation". The competency is inevitably linked to its author in the action and can only be observed in this context. If we refer to [Le Boterf, 1994], a competency is a combination of resources to be mobilized in order to achieve a goal. Those resources can be social skills, know-how or knowledge. The competency is not directly

observable but can be deduced from the activities carried out by the person. Here, an activity is a coherent set of completed tasks organized toward a predefined objective with a result that can be measured. An activity is formalized by using action verbs (control, perform, produce,...) and can be divided in sub activities or processes. Its description reveals the importance of the activity and the complexity of the tasks to achieve. Most of the time, a job offer describes in detail what activities should be carried out. That's why we decided to look for activities in job offers description to deduce the required competencies. Extract activities in job offers is about mining specific part of sentences in a web page which is a semi-structured textual environment.

Many papers approached competencies with a data mining point of view, to extract them from a large amount of data. The most common way to tackle this issue is the creation of competency ontologies [Amourache et al., 2008], [Harzallah et al., 2002], [Ziebarth et al., 2009]. This method has shown good results in those three articles, but suffers from the need of user's implication. Indeed, the creation of an ontology always requires an expert in the field to manage it and keep it up to date. With this in mind, it is not realistic to consider the use of an ontology at a large scale like in our context of detecting emerging competencies in a job offers website. Those papers rather agree on the definition of competency and only differs in terminology. We focus our work on activities in job offers but all the proposed methods rely on the competency itself regardless its source (Curriculum Vitae, job offer, ...). However competencies are not simple to observe, especially with those sources which are not reliable to directly look at competencies, and this shortcut can lead to many errors.

B. Emerging pattern

1) Pattern Mining: Pattern [Agrawal and Srikant, 1995] mining extraction is done by selecting patterns with a support above a threshold manually chosen [Agrawal and Srikant, 1994]. Two main algorithms were developed to mine patterns based on their frequency, Apriori [Agrawal and Srikant, 1994] and Eclat [Zaki, 2000]. The first one, is the most frequent pattern mining approach. Apriori is based around the monotonic property "All non empty itemsets of a frequent itemset must be frequent" [Agrawal et al., 1993]. It is a breadth first algorithm, it has to make multiple horizontal scans over the entire database. This leads to two cases of poor performances, if the database is very large and if the patterns' size is very long. Eclat is a depth first algorithm, which means it scans patterns vertically.

Our context gives us a contiguous constraint due to activities' nature. A contiguous pattern of n words is called an n-gram. [Fürnkranz, 1998] developed an algorithm based on Apriori to mine n-gram with a frequency above a threshold.

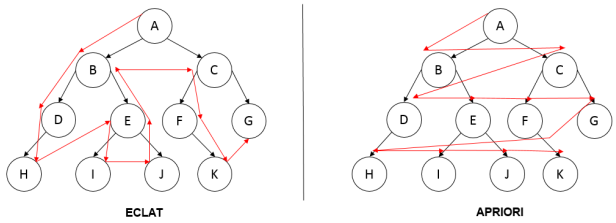

Figure 1. Frequent pattern mining

Algorithm 1 Ngram-mining(T,$MaxNGramSize$,$minsup$)

Input: T: corpus, *MaxNGramSize* : Integer, *minsup* : Integer
Output: $AllNGram$

```
Start
AllNGram[0] = {∅}
for n=1..MaxNGramSize
    Candidats = ∅
    AllNGram[n] = ∅
    foreach t ∈ T do
        foreach NGram ∈ NGram(t, n) do
            InitialGram = NGram − LastWord(NGram)
            FinalGram = NGram − FirstWord(NGram)
            if InitialGram ∈ AllNGram[n − 1]
            and FinalGram ∈ AllNGram[n − 1]then
                Count{NGram} = Count{NGram} + 1
                Candidats = Candidats ∪ NGram
    foreach NGram ∈ Candidats do
        if Count{NGram} ≥ minsup then
            AllNGram[n] = AllNGram[n] ∪ NGram
return AllNGram
End
```

Despite the disadvantages seen above regarding Apriori, the use of this n-gram mining suits our requirements, activities are not very long patterns, and seems to fit activities mining. We have to adapt the n-gram mining algorithm to extract activities in our context.

2) Emergence: Emerging pattern has been introduced by [Dong and Li, 1999], a pattern is called emerging between two datasets if the ratio of its supports, in those datasets, is superior to a given threshold. This definition does not allow more than two datasets and does not take time into account. We can use emerging pattern between successive periods, but we will not have a global emergence's point of view.

[Thorleuchter et al., 2014] brought out a method to detect emerging weak semantic signals. Their approach to spot emergences is based on the number of documents where the semantic signal appears. They crawl the web at different points in time and check if the percentage of documents with the signal growths. If it is the case, the signal becomes a candidate for emergence. Whereafter, depending on the percentage's rise, the signal is declared strongly increasing, slowly increasing, or static. This method offers a valid

way to detect emergence, but it does not allow a great anticipation. The last step is too open to interpretation and should be a more statistical result. Emerging trend detection is the ability to detect the growth in interest of a topic area over time [Minh-Hoang Le, 2005]. This paper's goal was to detect emerging trends in a set of scientific articles by extracting six features from articles to construct a ranking in interest and utility. It mentions a method to make a trend's evaluation over time, by using a linear regression on all data points. But this evaluation is not a part of the experiments. The use of a linear regression to evaluate the trend is applicable to the prediction of emergence. Our model will apply it to n-gram mining in the context of activities.

III. Prediction of the emergence

Our methodology consists in three steps. First of all, select and transform the data with web and text mining. Then mine the activities. Finally predict the emergence of activities. The prediction of emergence requires to gather data at different points of time $(t_0, t_1, ..., t_n)$, in order to get multiple periods and detect future emergence (see Fig 2.).

A. Web and text mining

We specified earlier that we would focus on the activities and not on the competencies directly because it seems easier and more accurate to observe activities. An activity is described by a sentence with an action verb. The data selection is entirely linked to the choice of data source [Cooley et al., 1997]. In our case, it depends on the job offers website. After data has been collected, we need to transform it in order to allow activities extraction based on n-gram frequency mining. We will use the standard tokenization [Hotho et al., 2005] to separate each term and then discard what we call stop words [Voorhees, 1999]. Those words, like "a" or "the" are not useful to the meaning of a text. Due to the nature of an activity, we have to identify all verbs. Lemmatization [Hull and Grefenstette, 1996] prevents from losing words' nature (noun, verb) unlike stemmating [Porter, 1980] which cuts words and denatures them. We will apply a French lemmatizer [Namer, 2000] which maps verbs forms to the infinite tense and notify you the nature of the term. It does the same thing with nouns, and maps them to the singular form. With this process, we greatly reduce the size of the dictionary (set of all distinct terms). We get a list of sequences with lemmatized terms in their original order, ready to extract activities.

B. Activity mining

When all irrelevant words are removed, an activity is an action verb with two to four words to define it. Thus, an activity is recognizable with three to five words. We will use the n-gram frequency algorithm [Fürnkranz, 1998] detailed above, with n going from three to five. A minimum support of three is sufficient to keep emerging n-grams while removing useless n-grams. We should not encounter drop performance by scaling up, even if this algorithm is based on Apriori, because the maximum size of n is only five. We elaborate a list of action verbs based on verbs used to write activities in job descriptions. To reduce the amount of n-grams and keep potential activities, we will discard those with no action verb from our list. This filter is only possible because of the lemmatization, indeed all verbs are in infinite tense and can be compared to the list of action verbs. Other filters might be possible, depending on the data source, in any case we absolutely need to set a maximum support threshold in order to discard n-grams already very frequent. The last filter only keeps closed n-gram. An n-gram is closed if there is not any n-grams included in it and with the same support [Yan et al., 2003].

Once filtering and activities extraction are done, we will have a list of n-grams for each observed period $(t_0, t_1, ..., t_n)$.

C. Predicting emerging activities

We need enough periods to detect future emergence. We apply a linear regression [Bakin, 1999] to our list of n-grams' supports by period. Based on the slope's sign and the value of the coefficient of determination, we will be able to tell if an n-gram is a solid candidate to emerge. The coefficient of determination (R^2) is defined by:

$$R^2 = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \qquad (1)$$

Where y_i are real values, $\bar{y}$ is the arithmetic mean of y_i and $\hat{y}_i$ are regression's values.
The slope's sign determines if the n-gram's support is increasing or not. The coefficient of determination (1) reflects the accuracy of the regression and thus if we can rely on the regression. Thereby, if an n-gram has a regression with a positive sign and a coefficient of determination over 0.5, the regression is accurate enough to consider this n-gram as a potential future emerging n-gram. It is only at this point that we will manually examine, for the moment, the list of candidates to check which ones correspond to real activities and get a final list of potential future emerging activities.

D. Evaluation

Our goal being the prediction of emerging activities, we have to focus on finding a maximum of relevant activities about the emergence predicted. We will evaluate our model on the capacity to detect activities and the capacity to predict an emergence. To evaluate the extraction of activities, we will use standard measures to evaluate an information retrieval (IR) system. The information retrieval's goal is to select potential relevant documents in a set of documents in response to a request [Hearst, 1999]. The quality of an IR

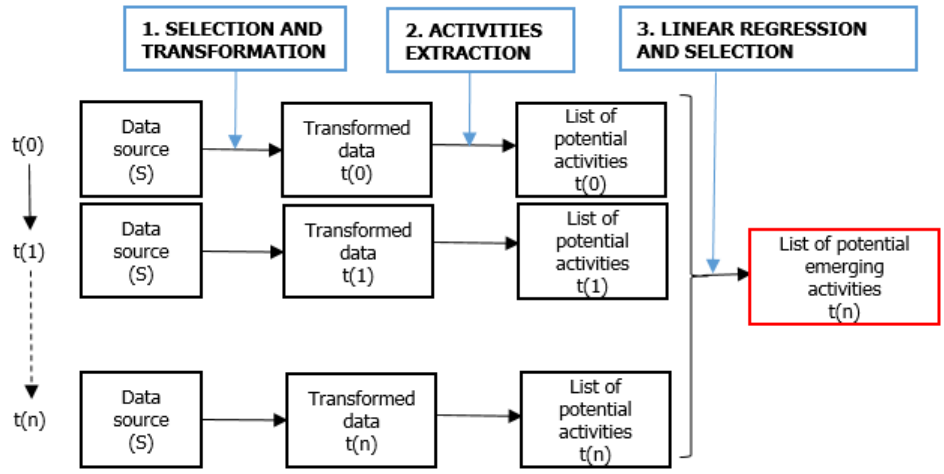

Figure 2. Processing of our model

system is determined by three measures: recall, precision and the $F_{measure}$ [Rijsbergen, 1979]. Recall is the number of documents that are relevant to the query that are successfully retrieved. Precision is the ratio between the number of relevant documents to the query retrieved and the number of documents retrieved. The $F_{measure}$ is the weighted harmonic mean of precision and recall:

$$F_{mesure} = \frac{2 \cdot R \cdot P}{R + P} \quad (2)$$

The $F_{measure}$ is a score, allowing to have an idea on the performance of an IR system. It measures the capacity of a system to give all the relevant information and to reject the others. We will use those three measures to evaluate our activity mining system. As we do not own a corpus with activities already identified, we will use some job offers and ask to an expert in the field to identify activities. Whereafter, we will be able to calculate recall, precision and the $F_{measure}$ to determine the quality of our system.

To evaluate the prediction of emergence, we need a textual corpus of pattern evolving through time and from which we already know emerging patterns. Then we can calculate recall, precision and the $F_{measure}$ based on emerging patterns retrieved by our system.

IV. Experiment

In this section, we present an experiment based on the model we just introduced. In France, the online employment market is quite abundant. In 2013, 13 000 job offers websites were counted. We chose one due to its accessibility, no need to register, but also due to its quality (we only keep jobs for executives) and quantity (around 40 000) of job offers. Each job is freely described by each company besides the job's title, the location, the wages, the company name and other pieces of information depending on the job offer. The description usually starts with the company description, then job's tasks and activities, and finishes with the applicant profile.

A. Data selection and transformation

Our model covers job offers on the website and records the title of the offer, the date of issue, as well as the description [Thelwall, 2001]. We collected offers during almost three months, we thus chose to make periods of two weeks to reach a total of six periods. The date of issue allows us to distribute the offers to the appropriate period (see Table I).

For the rest of this section we will refer to period with (t_0),(t_1),(t_2),(t_3),(t_4) and (t_5). We found almost 10 000 new offers by period. For every job offer, every term is separated (words and punctuation marks) then compared with our stop words list. If the term is in the list, it is removed. Then we proceed to the French lemmatization of every term using the tool *TreeTagger*[1]. We obtain a list of lemmatized set of terms corresponding to the complete corpus of job offers we recorded.

B. Frequent n-gram mining

We stated (see III. B.) that the activities could be reduced to an n-gram going from three to five terms. It is not relevant to keep n-grams appearing only one or two times because it can involve errors or marginal cases, it is what we call noise. Thus we are going to detect n-grams having a support strictly above three with n going from three to five. Whereafter, we will apply successively three filters to these n-grams, to reduce their number. The first filter consists in keeping only n-grams containing a verb in our list of action verbs. The second filter is going to discard the n-grams containing at least three verbs, because after observation of the activities we noticed that if an n-gram

[1]http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

6/10 au 19/10 (t_1)	20/10 au 2/11 (t_2)	3/11 au 16/11 (t_3)	17/11 au 30/11 (t_4)	1/12 au 14/12 (t_5)	15/12 au 28/12 (t_6)	Total
9183	9848	9914	10837	8821	11323	59926

Table I
COLLECTED JOB OFFERS IN 2014

n-grams	(t_1)	(t_2)	(t_3)	(t_4)	(t_5)	(t_6)	Total	Decrease(%)
All	123089	152797	152890	165320	137815	176838	908749 (528642 distincts)	-
With action verb	24478	31530	31478	34634	28534	36623	187277	79
Max 2 verbs	23656	30358	30300	33428	27438	35320	180500	3
Closed	13404	16709	16679	18526	14733	19591	99642	45
In 3 periods	-	-	-	-	-	-	39278 (9559 distincts)	61
1 *sup*>10	-	-	-	-	-	-	9997 (2097 distincts)	75
slope>0	-	-	-	-	-	-	1317	37
$R^2 > 0.5$	-	-	-	-	-	-	272	79

Table II
N-GRAMS

presents more than three verbs then it contains more than one activity. Finally, the last one keeps only closed n-grams (see III. B.).

After three filters, The total number of n-grams decreased of 89%. Only the filter number 2 (maximum two verbs) produces a low decrease of 3 %, for the rest the filters are very effective.

Now that we have a rather short list of n-grams, we are able to filter n-grams according to their supports and to their presence in each period. Our periods being rather short, we can not impose the presence of n-grams in each one of them. Thus we decided to keep n-grams present in at least half of total periods meaning three. This parameter can be modified according to the duration of the periods in which job offers are collected.

We are not going to limit the maximal value of the support to preserve the activities which are already frequent. However, we will only keep n-grams having at least a support of ten in one of the periods, to remove n-grams having no chance to emerge. We chose ten because any increase of support below this number is not relevant according to our experimental analysis of supports in our context.

These last two filters have allowed us to strongly decrease the number of n-grams. Indeed, we finish with 9997 among which only 2097 are distinct. Hereby, we succeeded in decreasing from 528642 distinct n-grams to 2097, whether a decrease of 99.6 %. It is now necessary to select the candidates for potential emergence.

C. Regression

We use a linear regression between the support of n-grams and the period. So for every n-gram we search α and β such as: $y = \alpha + \beta x$ where y is the support and x, the number of the period. So β is the slope of our regression and its sign indicates if the curve increases or decreases. Thus, we are going to only keep the n-grams of which the slope of the linear regression is positive. It remains to determine if the regression is close enough to real values to be relied on and anticipate a future emergence. We keep the n-grams with a coefficient of determination above 0.5.

The linear regression effectively filters n-grams, we are now decreasing from 2097 to 272 which is almost 87 % of decrease. We got to a completely manageable number of n-grams for a manual selection.

D. Future emerging activities selection

After a first manual selection, made by an expert in competencies and concerning only the title, it remains 40 n-grams (see Table III). We are now going to analyze the n-grams to get to our final list. First, we analyze the supports of every n-grams to rule out those who are clearly not in a case of future emergence. We need to remove n-grams already very frequent but we want to keep n-grams frequent or almost frequent. We calculate the mean support of all n-grams and only focus on the n-grams above. Then we take the mean support of those n-grams, add the standard deviation and use this value as limit. This rule removes four n-grams. We notice that several n-grams are coming from the same activity, we merge the concerned n-grams and it discards fourteen. We finally get a list of 22 potential emerging activities.

V. EVALUATION

For the evaluation of the prediction of emergence part, due to the lack of adapted corpus, we artificially created a corpus of hundred and twenty job offers distributed over four periods. We created an emergence by placing the same job offer, three times in the first period, four times in the second, eight times in the third and finally twelve times in

ID	n-grams	(t_1)	(t_2)	(t_3)	(t_4)	(t_5)	(t_6)	slope	R2	average support
1	faire evoluer rayon autonomie rester	3	7	6	15	20	23	4.2	0.93	12.33
2	hygiene securite faire evoluer rayon	3	7	6	15	20	23	4.2	0.93	12.33
3	norme regle hygiene securite faire	3	7	6	15	20	23	4.2	0.93	12.33
4	regle hygiene securite faire evoluer	3	7	6	15	20	23	4.2	0.93	12.33
5	securite faire evoluer rayon autonomie	3	7	6	15	20	23	4.2	0.93	12.33
6	optimiser organisation gestion rayon assure	4	7	6	15	20	24	4.2	0.93	12.67
7	mission crescendo recruter compte client	6	7	12	13	13	14	1.68	0.85	10.83
8	etendre systeme embarquer passer securite	-	-	-	5	13	12	3.0	0.82	10
9	rechercher solution technique	-	4	4	5	5	11	1.7	0.79	5.80
10	creer proposer large panel metier	-	9	4	10	11	18	2.9	0.78	10.40
11	consultant realiser action achat concret	4	3	3	6	8	12	1.7	0.78	6
12	etre consultant realiser action achat	4	3	3	6	8	12	1.7	0.78	6
13	realiser action achat concret pratique	4	3	3	6	8	12	1.7	0.78	6
14	competence capitaliser savoir faire domaine	-	-	4	3	23	22	5.1	0.77	13.25
15	centre competence capitaliser savoir faire	-	-	5	3	24	22	5.14	0.76	13.50
16	manager centre competence capitaliser savoir	-	-	5	3	24	22	5.14	0.76	13.50
17	appuyer support manager centre competence	-	-	4	3	24	22	5.2	0.76	13.25
18	developper client prestigieux secteur activite	-	10	4	10	11	18	2.8	0.73	10.60
19	proposer large panel metier complementaire	-	10	4	10	11	18	2.8	0.73	10.60
20	faire conception realisation operation compter	-	-	5	-	10	13	2.6	0.71	9.33
				...						
40	gerer projet client	-	1	2	1	3	23	3.4	0.52	6

Table III
N-GRAMS

the last one. Thus the emergent activities to be predicted are the activities in this job offer.

After application of our model on this corpus, we obtain a list of twenty three n-grams candidate to the emergence. All these n-grams arise from the emergent job offer, they do not correspond to twenty three activities in this offer. This number is due to the activities of the company publishing the offer. We thus obtain a success rate of 100 % between the real emergence and the predicted emergence. This result is to be tempered, because this evaluation would require to be repeated on a real corpus of which the emergences would already be known.

The real corpus of our study will be usable when we will have more than six periods and the validation on the activities which really emerged. Indeed, once we will clearly identify and validate emerging activities we will be able to perform the evaluation over the periods preceding the periods when the identified activities emerged.

Now we are going to evaluate the activity detection part of our model with the protocol we previously defined. One expert looked for the activities in twenty job offers we recorded. He found 184 activities. We applied our methods of detection to the same job offers with the following results:

Correct activities	92
Companies activities	291
Wrong activities	229
Rappel	$\frac{92}{184} = 0.5$
Precision	$\frac{383}{612} = 0.63$
$F_{measure}$	0.56

Table IV
MEASURE OF ACTIVITY DETECTION

Our model does not yet make the distinction between job activities and company activities. We decided to keep company activities to check the emerging one among them. However they can be count in the precision because they are activities after all. We can see that our method is quiet successful, despite the imprecise nature of an activity and the lack of structure of the data source. Our model can be improved by feeding our action verbs list with new encountered activities.

CONCLUSION

The new technologies democratized the culture of the immediacy creating a huge need for reactivity and anticipation. That is why notions such as the emergence became crucial to make better decisions at the right time. There already are many tools to detect almost or already established emergences. On the other hand, few works [Thorleuchter et al., 2014] dealt with the prediction of the

emergence. This paper puts forward a model of prediction of emerging patterns applied to activities on the market of online employment.

Our model allows on one hand, to detect the activities from a textual mass of data, and on the other hand, to establish candidates for a potential emergence over time. We showed that it was possible to lean on the n-gram mining to detect activities in a textual flow. The approach to predict the emergence of patterns by using a regression is promising, as shown by the results we obtained. The candidates to emerge are likely, but the model has to be tried on a longer period than three months, especially in the domain of the employment where the competencies take time to emerge.

The implementation of the model highlighted many suggested improvements and explorations. Our experiment revealed that it was possible to detect the activities from a text without additional information. It would also be very profitable if the system learnt automatically the verbs of action which are not in the list. We used an algorithm of n-gram mining because of the constraint of contiguity, but it would be interesting to use an algorithm of pattern mining with the possibility of having a step of one or two words to detect frequent patterns which are already known. The emergence part may be improved by using more complex regressions, but also by weighting supports according to the period in which they are. We could then give a bigger importance to the periods as they are closer to the present. The notion of emergence is closely linked to the notion of mutation. To go further in the approach of prediction, it would be necessary to be able to determine the patterns which are mutated to emerge in new patterns.

REFERENCES

[Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, pages 207–216, New York, NY, USA. ACM.

[Agrawal and Srikant, 1994] Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA. IEEE Computer Society.

[Amourache et al., 2008] Amourache, F., Boufaïda, Z., and Yahiaoui, L. (2008). Construction d'une ontologie basée compétence pour l'annotation des CVs/Offres d'emploi. In *10th Conference on Software Engineering and Artificial Intelligence (MCSEAI),Maghrebian Conference on Information Technologies (28-30 april)*, pages 1–7.

[Bakin, 1999] Bakin, S. (1999). Adaptive regression and model selection in data mining problems.

[Chen et al., 1996] Chen, M.-S., Han, J., and Yu, P. S. (1996). Data mining: An overview from a database perspective. volume 8, pages 866–883, Piscataway, NJ, USA. IEEE Educational Activities Department.

[Cooley et al., 1997] Cooley, R., Mobasher, B., and Srivastava, J. (1997). Web mining: Information and pattern discovery on the world wide web. In *ICTAI '97: Proceedings of the 9th International Conference on Tools with Artificial Intelligence*, page 558, Washington, DC, USA. IEEE Computer Society.

[Dong and Li, 1999] Dong, G. and Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 43–52, New York, NY, USA. ACM.

[Fayyad, 1996] (1996). Advances in knowledge discovery and data mining. Menlo Park, CA, USA. American Association for Artificial Intelligence.

[Fürnkranz, 1998] Fürnkranz, J. (1998). A study using n-gram features for text categorization.

[Harzallah et al., 2002] Harzallah, M., Leclère, M., and Trichet, F. (2002). Commoncv: modelling the competencies underlying a curriculum vitae. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering, SEKE 2002, Ischia, Italy, July 15-19, 2002*, pages 65–71.

[Hearst, 1999] Hearst, M. A. (1999). Untangling text data mining. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 3–10, College Park, Maryland, USA. Association for Computational Linguistics.

[Hotho et al., 2005] Hotho, A., Nürnberger, A., and Paaß, G. (2005). A brief survey of text mining. volume 20, pages 19–62.

[Hull and Grefenstette, 1996] Hull, D. A. and Grefenstette, G. (1996). A detailed analysis of english stemming algorithms.

[Le Boterf, 1994] Le Boterf, G. (1994). De la compétence. essai sur un attracteur étrange. page 176, Paris. Les Editions d'Organisation.

[Minh-Hoang Le, 2005] Minh-Hoang Le, Tu-Bao Ho, Y. N. (2005). Detecting Emerging Trends from Scientific Corpora.

[Namer, 2000] Namer, F. (2000). FLEMM: un analyseur flexionnel du français à base de règles. *Traitement Automatique des Langues*, 41(2):523–547.

[Porter, 1980] Porter, M. (1980). An algorithm for suffix stripping. volume 14, pages 130–137.

[Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). Information retrieval. Newton, MA, USA. Butterworth-Heinemann.

[Thelwall, 2001] Thelwall, M. (2001). A Web crawler design for data mining. volume 27, pages 319–325.

[Thorleuchter et al., 2014] Thorleuchter, D., Scheja, T., and den Poel, D. V. (2014). Semantic weak signal tracing. volume 41, pages 5009 – 5016.

[Tufféry, 2005] Tufféry, S. (2005). Data mining et statistique décisionnelle : l'intelligence dans les bases de données. Paris. Ed. Technip.

[Voorhees, 1999] Voorhees, E. (1999). Natural language processing and information retrieval. In *Information Extraction: Towards Scalable, Adaptable Systems*, pages 32–48. Springer.

[Wittorski, 1998] Wittorski, R. (1998). De la fabrication des compétences. volume 135, pages 57–69. Paris : Documentation francaise.

[Xu et al., 2011] Xu, G., Zhang, Y., and Li, L. (2011). Web content mining. In *Web Mining and Social Networking*, volume 6 of *Web Information Systems Engineering and Internet Technologies Book Series*, pages 71–87. Springer US.

[Yan et al., 2003] Yan, X., Han, J., and Afshar, R. (2003). Clospan: Mining closed sequential patterns in large datasets. In *In SDM*, pages 166–177.

[Zaiane et al., 1998] Zaiane, O. R., Han, J., Li, Z.-N., and Hou, J. (1998). Mining multimedia data. In *Proceedings of the 1998 Conference of the Centre for Advanced Studies on Collaborative Research*, CASCON 98, page 24. IBM Press.

[Zaki, 2000] Zaki, M. J. (2000). Scalable algorithms for association mining. volume 12, pages 372–390, Los Alamitos, CA, USA. IEEE Computer Society.

[Ziebarth et al., 2009] Ziebarth, S., Malzahn, N., and Hoppe, H. U. (2009). Using data mining techniques to support the creation of competency ontologies. In *AIED*, page 223–230. 00009.