Introductory experiments with evolutionary optimization of reflective semantic vector spaces

Daniel Devatman Hromada[1, 2]
(1) ChART, Université Paris 8, 2, rue de la Liberté 93526, St Denis Cedex 02, France
(2) URK, FEI STU, Ilkovičova 3, 812 19 Bratislava, Slovakia
hromi@giver.eu

Résumé. Task 4 of DEFT2014 was considered to be an instance of a classification problem with opened number of classes. We aimed to solve it by means of geometric mesurements within reflective vector spaces – every class is attributed a point C in the vector space, N document-denoting nearest neighbors of C are subsequently considered to belong to class denoted by C. Novelty of our method consists in way how we optimize the very construction of the semantic space: during the training, evolutionary algorithm looks for such combination of features which yields the vector space most « fit » for the classification. Slightly modified precision evaluation script and training corpus gold standard, both furnished by DEFT organiers, yielded a fitness function. Only word unigrams and bigrams extracted only from titles, author names, keywords and abstracts were taken into account as features triggering the reflective vector space construction processses. It is discutable whether evolutionary optimization of reflective vector spaces can be of certain interest since it had performed the partitioning of DEFT2014 testing corpus articles into 7 and 9 classes with micro-precision of 25%, respectively 31.8%.

Keywords: reflective semantic indexing, evolutionary optimization, opened class classification

1 Introduction

We understood the Task 4 of 2014 edition of the datamining competition Defi en Fouille Textuelle (DEFT) as an instance of multiclass classification problem. More concretely, the challenge was to create an artificial system which would be able attribute a specific member of the set of all class labels to scientific articles of the testing corpus. The training corpus of 208 scientific articles presented in diverse sessions of diverse editions of an annual TALN/RECITAL conference was furnished to facilitate the training of the model.

The tricky aspect of the challenge was, that one could be potentially asked, in the testing phase, to attribute to an object, which was not present during training phase, a label which was also not present in the turing phase.

For this reason, we had considered Task 4 to be an instance of an open-class variant of classification problem, i.e. a multiclass classification problem when one does not know in advance neither the number nor even the nature of categories which are to be constructed. We had decided to try to solve the problem of open-classification problem by a following approach, based principially on mutually intertwined notions of « object » and « feature » :

1. During the (train|learn)ing phase, use the training corpus to create a D-dimensional semantic vector space, i.e. attribute the vectors of length D to all members of the set of entities (word fragments, words, documents, phrases, patterns) E which includes all observables within the training corpus

2. During the testing phase:

2.1 characterize the object (text) O by a vector $\vec{o}$ calculated as a linear combination of vectors of features which are observable in O and whose vectors were learned during the training phase

2.2 characterize labels-to-be-attributed L1, L2, ... by vectors $\vec{l}_{1,} \vec{l}_2$

2.3 associate the object O with the closest label. In case we use cosine metric, we minimize angle between document vector and label vectors i.e. $argmax \cos\left(\vec{o}, \vec{l}_x\right)$

2 Evolutionary optimization of reflective vector spaces

Our learning algorithm consists of two nested components. The inner component is responsable for construction of the vector space. Its input is a genotype, the list of D features which trigger the whole reflective process, its output -a phenotype - is a D-dimensional vector space consisting of vectors for all features, objects (documents) and classes. The inner component is « reflective » in a sense that it multi-iteratively not only characterizes objects in terms of their associated features, but also features in terms of associated objects.

The envelopping outer component is a trivial evolutionary algorithm responsible whose task is to find the most « fit » combination of features to perform the classification task. In every « generation », it injects multiple genomes into the inner component and subsequently evaluates the fitness function of resulting vector spaces. It subsequently mutates, selects and crosses-over genotypes which had yielded the vector spaces wherein the classification was most precise.

2.1 Features

A feature is a concrete instance of an observable associated to a certain concrete object. In a text-mining scenarios, features are most often strings of characters. We had extracted two types of features : semantic and shallow.

2.1.1 Semantic features

Semantic features are tokens which, with very high probability, carry an important semantic information. Semantic features were extracted only from titles, author names, keywords and abstracts, since these pieces of content are considered to be semantically very dense. More concretely, all above mentioned elements were split into tokens with regular expression /[\W \n_]/, i.e. all non-word characters and newline played the role of token separators. Subsequently, every individual token which was not in PERL's Lingua::Stopwords[1] list was considered to be a separate feature. Also, in case of titles and keywords, couples of subsequent tokens were also considered as a feature. Note that fulltext versions of the articles were not considered as source of semantic features.

Pool of 5849 distinct semantic feature types, observable within at abstracts, titles, keywords or authornames of at least two distinct documents was extracted. Randomly chosen members of this pool have subsequently served as first genes triggering the construction of individual vector spaces.

2.1.2 Shallow features

Shallow, or surface features are features whose semantic information content is discutable, nonetheless they could potentially play the role of a useful classification clue. We have principially used the fulltexts of articles as a source of such features – all word 1-grams. 2-grams and 3-grams present in the fulltext were considered to be shallow features of class C, under the condition that they had occured only within two or more documents of the training corpus associated to class C.

During training, 2790 features were observed which occured in fulltexts of two (SF_{2+}) or more documents of the same class C and 160 features occured in fulltexts of three (SF_{3+}) or more documents of the same class C. If ever such features were observed in the document D of the testing corpus, the cosine between D and class C was increased with value of 0.02 to yield the final score.

[1] This list of stopwords was the only external resource used.

2.2 Reflective space indexing

We define as reflective a vector space containing both objects (documents) and their associated features fulfilling the circular condition that vectorial representations of objects (documents) are obtained by linear combinations of vectors of their features and vectors of features are obtained as as linear combinations of vectors of objects within which the feature occurs. Such a circularity - whereby objects are defined by features which are defined by objects which are … ad convergence - is considered as unproblematic and is, in fact, a wanted attribute of the space.

Thus, in a reflective model, both features and objects are members of the same D-dimensional vector space and can be represented as rows of the same matrix. Note that this is not the case in many existing vector space models whereby features (words) and objects (documents) are often represented either as elements of distinct matrices or, as columns, respectively rows of the same co-occurrence matrix.

A prominent model where such « entity comesurability » is assured is Reflective Random Indexing (Cohen et al., 2010) and had been the core component of the approach which had obtained particular performances in DEFT's 2012 edition (ElGhali et al., 2012).

The reflective space indexing (RSI) algorithm which we had deployed in this edition of DEFT is, in certain sense, a non-stochastic variant of RRI. It is non-stochastic in a sense that instead of randomly projecting huge amount of feature-concerning knowledge upon the space of restricted dimensionality, as RRI does, the algorithm rather departs from a restricted number of selected features which subsequently « trigger » the whole process of vector space construction.

RSI's principal parameter is the number of dimensions of the resulting space (D). Input of RSI is a vector of length D whose D elements denote D « triggering features », the initial conditions to which the algorithm is sensible in the initial iteration. After the algorithm has received such an input, it subsequently characterizes every object O (document) by a vector of values which represent the frequency of triggering feature in object O. Initially, every document is thus characterized as a sort of bag-of-triggering-features vector. Subsequently, vectors of all features – i.e. not only triggering ones – are calculated as a sum of vectors of documents within which they occur and a new iteration can start. In it, initial document vectors are discarded and new document vectors are obtained as a sum of vectors of features which are observable in the document. Whole process can be iterated multiple times until the system converges to stationary state, but it is often the second and third iteration which yields most interesting results. Note also that what applies for features and objects applies, *mutatis mutandi*, also for class labels.

For purposes of DEFT 2014, every individual RSI run consisted of 2 iterations and yielded 200-dimensional space.

2.3 Evolutionary optimization

The evolutionary component of the system hereby introduced is a sort of feature selection mechanism. The objective of the optimization is to find such a genotype – i.e. such a vector of triggering features – which would subsequently lead to discovery of a vector space whose topology would construction of a most classification-friendly vector space.

As is common in evolutionary computing domain, whole process is started by creation of a random population of individuals. Each individual is fully described by a genome composed of 200 genes. Initially, every gene is assigned a value randomly chosen from the pool of 5849 feature types observable in the training corpus. In DEFT2014's Task 4 there were thus 5849^{200} possible individual genotypes one could potentially generate and we consider it important to underline that classificatory performance of phenotypes, i.e. vector spaces generated by RSI from genotypes, can also substantially vary.

What's more, our observations indicate that by submitting the genotype to evolutionary pressures -i.e. by discarding the least « fit » genomes and promoting, varying and replicating the most fit ones - one also augments the classificatory performance of the resulting phenotypical vector space. In other terms, search for a vector space[2] which is optimal in regards to subsequent partitioning or clustering can be accelerated by means of evolutionary computation.

During the training, evaluation of fitness of every individual in every generation proceeded in a following manner :

- pass the genotype as an input to RSI (D=200, I=2)
- within the resulting vector space, calculate cosines between all document and class vectors
- in case of use of shallow features adjust score accordingly (c.f. 2.1.2)
- attribute N documents with highest score to every class label (N was furnished for both testing and training corpus)
- calculate the precision in regards to training corpus golden standard. Precision is considered to be equivalent to individual's fitness

Size of population was 50 individuals. In every generation, after the fitness of all individuals has been evaluated, 40% of new individuals were generated from the old ones by means of a one-point crossover operator whereby the probability of the individual to be chosen as a parent was proportional to individual's fitness (Sekaj, 2005). For the rest of the new population, it was generated from the old one by combination of fitness proportionate selection and mutation occuring with 0.01 probability. Mutation was implemented as a replacement of a value in a genome by another value, randomly chosen in the pool of 5849 feature types.

Advanced techniques like parallel evolutionary algorithms or parameter auto-adaptation were not used in this study.

3 Results

The vector space VS_1 , which we had decided to use as a model for testing phase, was constructed by RSI triggered by the following genome:

ressource # premier # notions # 100 # agit # raisons # french # syntaxe # naturelles # conditionnels # fonctionnelle # adjoints # terminologie # permettre # paraphrases # filtrage # proposons # fois # perspectives # technique # expérience # wikipédia # 2 # arbres adjoints # selon # fonctionnalités # reste # sélection # filtrage # permettant # mesurer # lexiques # bleu # énoncés # couverture # intégrer # formel # transcriptions # décrit # absence # tant # notions # analyseur # delphine bernhard # montrent # aligner # faire # fournies # large # entité # simples # basées # faire # syntaxe # couples # distinguer # mesures # enfin # effet # amélioration # premiers # erreur # morphologique # 0 # formelle # bilingues # sélection # point # partie # consiste # paires # autre # enfin # étiquettes # valeur # surface # caractériser # vincent claveau # comment # élaboration # proposée # travail # bien # paralléles # bonnes # enrichissement # extraits # travail # adjoints # combiner # spécifiquement # nommées # basé # comparé # réflexion # nécessaire # ressource # résultat # lorsqu # montrent # segmenter # vise # avoir # statistiques # objet # mise # interface syntaxe # annotation # arabe # traduction automatique # lexiques bilingues # exemple # comparaison # autres # extraites # plusieurs # jeu # téche # traduction automatique # discursifs # nommées # phrase # fouille # constitué # événements # manque # formel # utilisateurs # initialement # présenté # semble # anglais # score # grande # cas # chaque # langue # interface # ci # mesurer # évaluons # originale # structures # générique # utilise # analyse syntaxique # arabe # travail # différents # franéaise # trés # wordnet # structure # enrichissement # noyau # donné # propriétés # énoncé # aléatoires # afin # exploite # développement # résoudre # générer # proposé # énoncé # elles # domaines # production # arbres # travail # régles # extraction information # textuels # morphologiques # fonctionnalités # modélisation # terme # syntaxe # compréhension # résultats # création # langage # représentation # étape # langues # représente # concluons # grandes # problématique # multi # absence # problématique # capable # telles # bonnes # abord # probléme # parole # représentation #

Run	Training	Testing
VS_1	0.87	0.2777
VS_1+SF_{2+}	0.99	0.2222
VS_1+SF_{3+}	0.98	0.2777

TABLE 1 : Average micro-precision of classification within VS_1 with/without use of shallow features

[2] A question may be posed : Why evolve the genotypic vector of triggering features and not directly the ultimate phenotypic vector space ? An answer could be : it is substantially less costly to optimize vectors than matrices. Nature does such « tricks » all the time.

4 Discussion

The algorithm hereby presented had attained the lowest result in Task 4 of DEFT2014 competition. When compared with other approaches - like that of ElGhali&ElGhali (this volume), that had attained an ideal 100% precision – it can be disregarded as strongly underperformant. It can be indeed true that the path of evolutionary optimization of reflective vector spaces is not a path to be taken by those linguists and engineers whose objective is to discover the best model for solving the minute task at hand, but only by those who strive for « something different ».

Failure notwithstanding, the approach briefly sketched in this article classified the data definitely better than a random process which indicates that it could be, at least potentially, useful. As other conceptors of novel approaches aiming two unite two disparate worlds – in our case the world of evolutionary computing with that of semantic vector spaces – we have been both confronted with huge amount of design choices and as such were prone to comitting implementation errors. In the case of our DEFT2014 tentative, we are aware of multiple mistakes: Primo, we had submitted as our DEFT2014 challenge contribution the test **data produced by a vector space trained in a scenario without any cross-validation**. It is evident that we have to pay the price for over-fitting. Secundo, we had stained two out of three runs with the « shallow features »; we should have rather focused on submitting runs based on other vector spaces. Discussion of other malchosen parameters and omissions – related to both reflective and evolutionary components of the algorithm - are beyond the scope of this article.

Also, it may still be the case that evolutionary optimization of vector spaces can be useful for solving the problems which are unsimilar DEFT2014's Task 4. In fact, we principially develop the model for the purpose of performance of computational modelization of both ontogeny and phylogeny of human linguistic competence. Our aim is principially to computationally simulate certain phenomena studied by developmental psycholinguistics or evolutionary psychology and to do it in a cognitively plausible (Hromada, 2014) way. The extent in which such models could be useful for solving somewhat cognitively implausible[3] text-mining tasks is a place for argument.

At last but not least, we consider that there is at least one contribution of our study which is not to be underestimated. That is: «a trivial observation» that by evolutionary selection of chromosome of features which initially « trigger » the reflective process one can, indeed, optimize the topology and hence the classification performance of the resulting vector space.

Acknowledgments

We would like to thank doc. Ivan Sekaj and technicians of Slovak University of Technology's FEI-URK department for initiation into Matlab clustering mysteries; to prof. Charles Tijus and Adil ElGhali for their moral support and to DEFT2014 organizers for a stimulating challenge.

References

COHEN T., SCHVANEVELDT R., WIDDOWS D. (2010). Reflective Random Indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, *43*(2), 240-256.

ELGHALI A., HROMADA D., ELGHALI K. Enrichir et raisonner sur des espaces sémantiques pour l'attribution de mots-clése la communication. Actes de *JEP-TALN-RECITAL*, 77.

HROMADA D. D. (2014). Conditions for cognitive plausibility of computational models of category induction. Accepted for conference *15TH INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING AND MANAGEMENT OF UNCERTAINTY IN KNOWLEDGE-BASED SYSTEMS*.

SEKAJ I. (2005). *Evolučné výpočty a ich využitie v praxi*. Bratislava : Iris.

[3] Only rarely is one, in real life, confronted with the task to classify X objects into N classes in a way that cardinality of classes-to-be-constructed is known in advance.