Trainable Tree Distance and an application to Question Categorisation*

Martin Emms
School of Computer Science and Statistics
Trinity College, Dublin
Ireland
Martin.Emms@tcd.ie

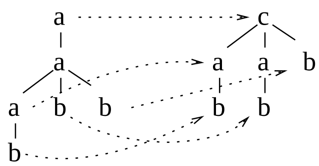
Abstract

Continuing a line of work initiated in Boyer et al. (2007), the generalisation of stochastic string distance to a stochastic tree distance, specifically to stochastic Tai distance, is considered. An issue in modifying the Zhang/Shasha tree-distance algorithm to the stochastic variants is noted, a Viterbi EM cost-adaptation algorithm for this distance is proposed and a counter-example noted to an all-paths EM proposal. Experiments are reported in which a k-NN categorisation algorithm is applied to a semantically categorised, syntactically annotated corpus. We show that a 67.7% base-line using standard unit-costs can be improved to 72.5% by cost adaptation.

1 Theory and Algorithms

The classification of syntactic structures into semantic categories arises in a number of settings. A possible approach to such a classifier is to compute a category for a test item based on its distances to a set of k nearest neighbours in a pre-categorised example set. This paper takes such an approach, deploying variants of *tree-distance*, a measure which has been used with some success in tasks such as Question-Answering, Entailment Recognition and Semantic Role Labelling (Punyakanok et al., 2004; Kouylekov and Magnini, 2005; Emms, 2006a; Emms, 2006b; Franco-Penya, 2010). An issue which will be considered is how to *adapt* the atomic costs underlying the tree-distance measure.

Tai (1979) first proposed a tree-distance measure. Where S and T are ordered, labelled trees, a *Tai* mapping is a *partial*, *1-to-1* mapping σ from the nodes of S to the nodes of T, which respects *left-to-right order* and *ancestry*[1], such as



A cost can be assigned to a mapping σ based on the nodes of S and T which are not 'touched' by σ, and the set of pairs (i, j) in σ. The *Tai*- or *tree-distance* $\Delta(S, T)$ is defined as the cost of the least-costly Tai mapping between S and T. Equivalently, tree-*edit* operations may be specified, and the distance defined by the cost of the least costly sequence of edit operations transforming S into T, compactly recorded as an edit-script:

operation	*edit-script element*
$m'(\vec{l}, \mathbf{m}(\vec{d}), \vec{r}) \to m'(\vec{l}, \vec{d}, \vec{r})$	(m, λ)
$m'(\vec{l}, \vec{d}, \vec{r}) \to m'(\vec{l}, \mathbf{m}(\vec{d}), \vec{r})$	(λ, m)
$\mathbf{m}(\vec{d}) \to \mathbf{m}'(\vec{d})$	(m, m')

An edit-script can be seen as a serialization of a mapping, and the distances via scripts and via mappings are equivalent (Zhang and Shasha, 1989).

If strings are treated as vertical trees, the Tai distance becomes the standard string distance (Wagner and Fischer, 1974). Ristad and Yianilos (1998) pioneered a probabilistic perspective on string distance via a model in which there is a probability distribution p on edit-script components, and $P(e_1 \ldots e_n) = \prod_i p(e_i)$. It is natural to consider how this probabilistic perspective can be applied to tree-distance, and the simplest possibility is to use

* This work was supported in part through the *Centre for Next Generation Localisation* by the Science Foundation Ireland

[1] so if (i_1, j_1) and (i_2, j_2) are in the mapping, then (T1) $left(i_1, i_2)$ iff $left(j_1, j_2)$ and (T2) $anc(i_1, i_2)$ iff $anc(j_1, j_2)$

exactly the same model of edit-script probability, leading to[2]:

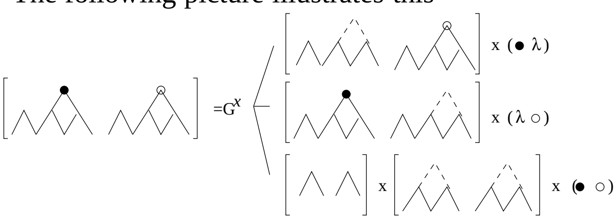
Definition 1.1 *(*All-paths and Viterbi stochastic Tai distance*)* $\Delta^A(S,T)$ *is the sum of the probabilities of all edit-scripts which represent a* Tai*-mapping from* S *to* T*;* $\Delta^V(S,T)$ *is the probability of the most probable edit-script*

Computing Δ^A and Δ^V We have adapted the Zhang/Shasha algorithm for Tai-distance to the stochastic case. The algorithm operates on the left-to-right post-order traversals of trees[3]. If i is (the index of) a node of the tree, let $\gamma(i)$ be its label, i_l be the leaf reached by following the left-branch down, and $S[i]$ be the sub-tree of S rooted at i. If i' is a member of $S[i]$, the prefix $i_l..i'$ of the traversal of $S[i]$ can be seen as a *forest* of subtrees. Considering the mappings between such forests, a case distinction can made on the possible final element of any script serializing the mapping, giving the following decomposition for the calculation of Δ^V and Δ^A

Lemma 1.1 *where* G^V *is the* max *operation, and* G^A *is the* sum *operation, for* $x \in \{V, A\}$

$$\Delta^x(i_l..i', j_l..j') = G^x \begin{cases} \Delta^x(i_l..i'-1, j_l..j') \times p(\gamma(i'), \lambda) \\ \Delta^x(i_l..i', j_l..j'-1) \times p(\lambda, \gamma(j')) \\ \Delta^x(i_l..i_l'-1, j_l..j_l'-1) \times \\ \quad \underbrace{\Delta^x(i_l'..i'-1, j_l'..j'-1) \times p(\gamma(i'), \gamma(j'))}_{\Delta^x_M(i_l'..i', j_l'..j')} \end{cases}$$

The following picture illustrates this



For any leaf i, the highest node k such that $i = k_l$ is a *key-root*, and $KR(S)$ is the key-roots of S ordered by post-order traversal. For $x \in \{A, V\}$, the main loop of TD^x is then essentially

for $i \in KR(S)$, $j \in KR(T)$
for $i' : i_l \le i' \le i$, $j' : j_l \le j' \le j$,
compute $\Delta^x(i_l..i', j_l..j')$ *via Lemma 1.1*

computing a series of forest distance tables, whilst reading and updating a persistent tree table. Space precludes further details except to note the subtlety in TD^A that to avoid double counting, the tree table must store values only for mappings between trees with matched or subsituted roots (the $\Delta^A_M(i_l'..i', j_l'..j')$ term in Lemma 1.1), unlike the Zhang/Shasha algorithm, where it stores the true tree-distance[4].

TD^A and TD^V work under a negated logarithmic mapping[5], with $\times/max/sum$ mapped to $+/min/logsum$[6]. Where Σ is the label alphabet, a *cost table* $\mathcal{C}$ of dimensions $(|\Sigma|+1) \times (|\Sigma|+1)$ represents (neg-logs of) atomic edit operation, with first column and row for deletions and insertions. For Δ^V and Δ^A, the probabilities represented in $\mathcal{C}$ should sum to 1. For TD^V, the neg-log mapping is never inverted and TD^V can be run with arbitrary $\mathcal{C}$ and calculates then the standard non-stochastic Tai distance. The *unit-cost* table, $\mathcal{C}_{01}$, has 0 on the diagonal and 1 everywhere else.

Adapting costs We are interested in putting tree-distance measures to work in deriving a category for an uncategorised item, using an *example-set* of categorised examples, via the k nearest-neighbour (kNN) algorithm. The performance of the kNN classification algorithm will vary with cost-table $\mathcal{C}$ and Expectation-Maximisation (EM) is a possible approach to setting $\mathcal{C}$. Given a corpus of training pairs, let the *brute-force all-paths EM algorithm*, EM^A_{bf}, consists in iterations of: **(E)** generate a virtual corpus of scripts by treating each training pair (S,T) as standing for the edit-scripts $\mathcal{A}$, which can relate S to T, weighting each by its conditional probability $P(\mathcal{A})/\Delta^A(S,T)$, under current costs $\mathcal{C}$ and **(M)** apply maximum likelihood estimation to the virtual corpus to derive a new cost-table. EM^A_{bf} is not feasible. Let EM^V be a Viterbi variant of this working with a virtual corpus of *best*-scripts only, effectively weighting each by the proportion it represents of the all-paths sum, $\Delta^V(S,T)/\Delta^A(S,T)$. Space precludes further details of EM^V. Such Viterbi training variants have been found beneficial, for example in the context of parameter training for PCFGs (Benedí and Sánchez, 2005). The training set for EM^V is tree pairs (S,T), where for each

[2] Δ^A was proposed by Boyer et al. (2007)

[3] so parent follows children

[4] Boyer et al. (2007) present somewhat unclear algorithms for Δ^A, not explicitly as extensions of the Zhang/Shasha algorithm, and do not remark this double-counting subtlety. Their on-line implementation (SEDiL, 2008) can compute incorrect values and this work uses our own implementation of the algorithms here outlined.

[5] $x = neg - log(p)$ iff $p = 2^{-x}$

[6] $logsum(x_1 \ldots x_n) = -log(\sum_i (2^{-x_i}))$

example-set tree S, T is a nearest same-category neighbour. EM^V increases the edit-script probability for scripts linking these trees, lessening their distance. Note that without the stochastic constraints on $\mathcal{C}$, the distance via TD^V could be minimised to zero by setting all costs to zero, but this would be of no value in improving the categorisation performance.

To initialize EM^V, let $\mathcal{C}_u(d)$ stand for a stochastically valid cost-table, with the additional properties that (i) all diagonal entries are equal (ii) all non-diagonal entries are equal (iii) diagonal entries are d times more probable than non-diagonal. As a *smoothing* option concerning a table $\mathcal{C}$ derived by EM^V, let $\mathcal{C}_\lambda$ be its interpolation with the original $\mathcal{C}_u(d)$ as follows

$$2^{-\mathcal{C}_\lambda[x][y]} = \lambda(2^{-\mathcal{C}[x][y]}) + (1-\lambda)(2^{-\mathcal{C}_u(d)[x][y]})$$

For stochastic string-distance Ristad and Yianilos (1998) provided a feasible equivalent to EM^A_{bf} : for each training pair (s,t), first *position-dependent* expectations $\mathcal{E}[i][j](x,y)$ are computed, then later summed into position-independent expectations. Boyer et al. (2007) contains a proposal in a similar spirit to provide a feasible equivalent to EM^A_{bf} but the proposal factorizes the problem in a way which is invalid given the ancestry-preservation aspect of Tai mappings[7]. For example, using a post-fix notation subscripting by post-order position, let $t_1 = (\cdot_1\ (\cdot_2\ \cdot_3\ m_4)\ \cdot_5\ \cdot_6)$, $t_2 = ((\cdot_1\ \cdot_2)\ (\cdot_3\ m'_4)\ (\cdot_5\ \cdot_6)\ \cdot_7)$ (from fig 3 of their paper). They propose to calculate a swap expectation $\mathcal{E}[4,4](m,m')$ by

$$[\Delta^A((\cdot_1),(\cdot_1\cdot_2)) \times [\Delta^A((\cdot_2)(\cdot_3),(\cdot_3)) \times p(m,m')] \times\Delta^A((\cdot_5\cdot_6),((\cdot_5\cdot_6)\cdot_7))]/\Delta^A(t_1,t_2)$$

But $\Delta^A((\cdot_5\cdot_6),((\cdot_5\cdot_6)\cdot_7))$ will contain contributions from scripts which map t_1's $\cdot_6$, an ancestor of m_4, to t_2's $\cdot_6$, a non-ancestor of m'_4, and these should not contribute to $\mathcal{E}[4,4](m,m')$.

2 Experiments

QuestionBank (QB) is a hand-corrected treebank for questions (Judge, 2006). A substantical percentage of the questions in QB are taken from a corpus of semantically categorised, syntactically unannotated questions (CCG, 2001). From these two corpora we created a corpus of 2755 semantically categorised, syntactically analysed questions[8], spread over the semantic categories as follows[9]: HUM(23.5%), ENTY(22.5%), DESC(19.4%), NUM(16.7%), LOC(16.5%) and ABBR(1.4%)

This corpus was used in a number of experiments on kNN classication using the tree-distance TD^V algorithm, with various cost tables. In each case 10-fold cross-validation was used with a 9:1 example-set/test-set split.

Figure 1 shows some results of a first set of experiments, with unit-costs and then with some stochastic variants. For the stochastic variants, the cost initialisation was $\mathcal{C}_u(3)$ in each case.

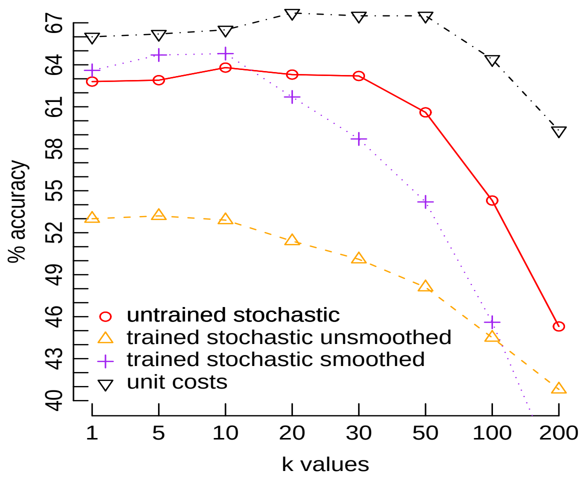


Figure 1: *Categorisation performance with unit costs and some stochastic variants*

The first thing to note is that performance with unit-costs (▽, max. 67.7%) exceeds performance with the non-adapted $\mathcal{C}_u(3)$ costs (○, max. 63.8%). Though not shown, this remains the case with far higher settings of the diagonal factor. Performance after applying EM^V to adapt costs (△, max. 53.2%) is worse than the initial performance (○, max. 63.8%). A Leave-One-Out evaluation, in which *example-set* items are categorised using the method on the remainder of the example-set, gives accuracies of 91% to 99%, indicating EM^V has made the best-scripts connecting the training pairs *too* probable, *over-fitting* the cost table. The vocabulary is sufficiently thinly spread over the training pairs that its quite easy for the learning algorithm to fix costs which make almost everything but exactly the training pairs have zero probability. The performance when smoothing is applied (+,max. 64.8%), interpolating the adapted costs with the initial cost,

[7] A fact which they concede p.c.

[8] available at www.scss.tcd.ie/Martin.Emms/quest_cat

[9] See (CCG, 2001) for details of the semantic category labels

with $\lambda = 0.99$, is considerably higher than without smoothing (△), attains a slightly higher maximum than with unadapted costs (○), but is still worse than with unit costs (▽).

The following is a selection from the top 1% of adapted swap costs.

8.50	?	.	12.31	The	the
8.93	NNP	NN	12.65	you	I
9.47	VBD	VBZ	13.60	can	do
9.51	NNS	NN	13.83	many	much
9.78	a	the	13.92	city	state
11.03	was	is	13.93	city	country
11.03	's	is			

These learned preferences are to some extent intuitive, exchanging punctuation marks, words differing only by capitalisation, related parts of speech, verbs and their contractions and so on. One might expect this discounting of these swaps relative to others to assist the categorisation, though the results reported so far indicate that it did not. A stochastically valid cost table cannot have zero costs on the diagonal, and even with a very high ratio between the diagonal and off-diagonal probabilities, the diagonal costs are not negligible. Perhaps this mitigates against success and invites consideration of outcomes if a final step is applied in which all the entries on the diagonal are zeroed. In work on adapting cost-tables for a stochastic version of *string distance* used in duplicate detection, Bilenko and Mooney (2003) used essentially this same approach. Figure 2 shows outcomes when the trained and smoothed costs finally have the diagonal zeroed.

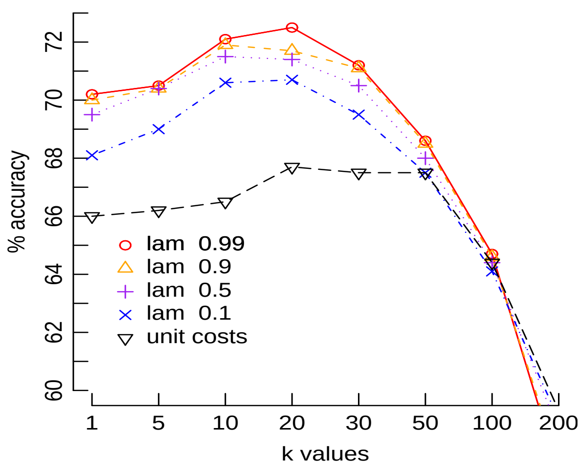


Figure 2: *Categorisation performance: adapted costs with smoothing and zeroing*

The (▽) series once again shows the outcomes with unit-costs whilst the other series show outcomes obtained with costs adapted by EM^V, smoothed at varius levels of interpolation ($\lambda \in \{0.99, 0.9, 0.5, 0.1\}$) and with the diagonal zeroed. Now the unit costs base-line is clearly outperformed, the best result being 72.5% ($k = 20$, $\lambda = 0.99$), as compared to 67.5% for unit-costs ($k = 20$)

3 Comparisons and Conclusions

Collins and Duffy (2001) proposed the $SST(S,T)$ tree-kernel 'similarity': a product in an infinite vector space, the dimensions of which are counts $c(t)$ of tree substructures t, each $c(t)$ weighted by a decay factor $\gamma^{size(t)}$, $0 < \gamma \leq 1$, and it has been applied to tree classification tasks (Quarternoni et al., 2007). If the negation of $SST(S,T)$ is used as an alternative to $\Delta^V(S,T)$ in the kNN algorithm, we found worse results are obtained[10], $64\% - 69.4\%$, with maximum at $k = 10$. However, deploying $SST(S,T)$ as a kernel in one-vs-one SVM classification[11], a considerably higher value, 81.3%, was obtained.

Thus, although we have shown a way to adapt the costs used by the tree-distance measure which improves the kNN classification performance from 67.7% to 72.5%, the performance is less than obtained using tree-kernels and SVM classification. As to the reasons for this difference and whether it is insuperable one can only speculate. The data set was relatively small and it remains for future work to see whether on larger data-sets the outcomes are less dependent on smoothing considerations and whether the kNN accuracy increases. The one-vs-one SVM approach to n-way classification trains $n(n-1)/2$ binary classifiers, whereas the approach described here has one cost adaptation for all the categories, and a possibility would be to do class-specific cost adaptation, in a fashion similar to Paredes and Vidal (2006).

One topic for future work is to consider how this proposal for cost adaptation relates to other recent proposals concerning adaptive tree measures (Takasu et al., 2007; Dalvi et al., 2009) as well as to consider cost-adaptation outcomes in some of the other areas in which tree-distance has been applied.

[10] using the SVMLIGHTTK (2003) implementation

[11] using the libsvm (2003) implementation, with decay $\gamma = 0.4$, slack $C = 2.0$

References

José-Miguel Benedí and Joan-Andreu Sánchez. 2005. Estimation of stochastic context-free grammars and their use as language models. *Computer Speech and Language*, 19(3):249–274, July.

Mikhail Bilenko and Raymond J. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pages 39–48.

Laurent Boyer, Amaury Habrard, and Marc Sebban. 2007. Learning metrics between tree structured data: Application to image recognition. In *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, pages 54–66.

CCG. 2001. corpus of classified questions by Cognitive Computation Group, University of Illinois `l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC`.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS14)*.

Nilesh Dalvi, Philip Bohannon, and Fei Sha. 2009. Robust web extraction: an approach based on a probabilistic tree-edit model. In *SIGMOD '09: Proceedings of the 35th SIGMOD international conference on Management of data*, pages 335–348, New York, NY, USA. ACM.

Martin Emms. 2006a. Clustering by tree distance for parse tree normalisation. In *Proceedings of NLUCS 2006*, pages 91–100.

Martin Emms. 2006b. Variants of tree similarity in a question answering task. In *Proceedings of the Workshop on Linguistic Distances, held in conjunction with COLING 2006*, pages 100–108, Sydney, Australia, July. Association for Computational Linguistics.

Hector-Hugo Franco-Penya. 2010. Edit tree distance alignments for semantic role labelling. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 79–84, Uppsala, Sweden, July. Association for Computational Linguistics.

John Judge. 2006. *Adapting and Developing Linguistic Resources for Question Answering*. Ph.D. thesis, Dublin City University.

Milen Kouylekov and Bernardo Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In Ido Dagan, Oren Glickman, and Bernardo Magnini, editors, *Proceedings of the first Pascal Recognising Textual Entailment Challenge Workshop*.

libsvm. 2003. library for svm `www.csie.ntu.edu.tw/~cjlin/libsvm`.

Roberto Paredes and Enrique Vidal. 2006. Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1100–1110.

Vasin Punyakanok, Dan Roth, and Wen tau Yih. 2004. Natural language inference via dependency tree mapping: An application to question answering. *Computational Linguistics*.

Silvia Quarternoni, Alessandro Moschitti, Suresh Manandhar, and Roberto Basili. 2007. Advanced structural representations for question classification and answer re-ranking. In *Advances in Information Retrieval, proceedings of ECIR 2007*. Springer.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(5):522–532, May.

SEDiL. 2008. Software for computing stochastic tree distance `http://labh-curien.univ-st-etienne.fr/informatique/SEDiL`.

SVMLIGHTTK. 2003. tree-kernel software `disi.unitn.it/moschitti/Tree-Kernel.htm`.

Kuo-Chung Tai. 1979. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):433.

Atsuhiro Takasu, Daiji Fukagawa, and Tatsuya Akutsu. 2007. Statistical learning algorithm for tree similarity. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 667–672, Washington, DC, USA. IEEE Computer Society.

Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173, January.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, 18:1245–1262.