

# Visual Exploration of Classifiers for Hybrid Textual and Geospatial Matching

Harald Sanftmann, Andre Blessing, Hinrich Schütze, Daniel Weiskopf

VIS, IfNLP, IfNLP, VISUS – Universität Stuttgart, Germany

Email: [sanftmann@vis.uni-stuttgart.de](mailto:sanftmann@vis.uni-stuttgart.de) [blessing@ims.uni-stuttgart.de](mailto:blessing@ims.uni-stuttgart.de)  
[schuetze2009@ifnlp.org](mailto:schuetze2009@ifnlp.org) [weiskopf@visus.uni-stuttgart.de](mailto:weiskopf@visus.uni-stuttgart.de)

## Abstract

The availability of large geospatial data from different sources has dramatically increased, but for the usage of such data in geo-mashup or context-aware systems, a data fusion component is necessary. To solve the integration issue classifiers are obtained by supervised training, with feature vectors derived from textual and geospatial attributes. In an application example, a coherent part of Germany was annotated by humans and used for supervised learning. Annotation by humans is not free of errors, which decreases the performance of the classifier. We show how visual analytics techniques can be used to efficiently detect such false annotations. Especially the textual features introduce high-dimensional feature vectors, where visual analytics becomes important and helps to understand and improve the trained classifiers. Particular technical components used in our systems are scatterplots, multiple coordinated views, and interactive data drill-down.

## 1 Introduction

This application paper is studying the usefulness of visual analytics for designing, debugging, and optimizing machine-learning techniques in geospatial text-based matching. Our group consists of experts in visualization and interactive systems, together with machine-learning and natural language processing (NLP) experts, and is embedded in an interdisciplinary research project that covers areas of computer science, electrical engineering, and philosophy related to context-aware information systems.

Context-aware systems adapt to changing environmental conditions, but they need information of

the environment. The NEXUS [4] platform provides data for context-aware systems by representing the real world through an internal *world model*. Data in the world model is heterogeneous and unstructured. Some data items are geo-referenced or geometric, others are often complex textual representations. For an open system like NEXUS it is obligatory to handle data from several data providers, making automatic data integration an important task. In general, these data sets are not equally modeled and describe different – although largely overlapping – sets of objects. Approaches from the natural language processing and machine learning (ML) communities can be combined to address these new issues. However, the complex mechanisms in NLP and ML are hard to understand and error analysis and system optimization are not easy. We use visual exploration to facilitate these tasks.

In this work we focus on the aggregation of geospatial data from different providers. In particular, each provider may have different measurements on which they base the model, resulting in data quality that varies from provider to provider. Our goal is to create a homogeneous model out of underlying heterogeneous models. A central issue is to find coherent instances in the different data sets.

We consider data sets from two exemplary providers, covering the same geospatial area. The data sets do not have the same schema, and use different attributes. Interestingly, there are also intentional differences between the data sets; e.g., providers include fictitious objects used as watermarks. Data sets also differ in quantity due to variations in density and types of geo-objects modeled. Our approach is to first create small training sets – consisting of coherent and incoherent instances of the two providers – manually and then to train a model that delivers correct coherence pairs for the

whole data sets. After this step it will be possible to perform an assessment of consistency and completeness of the two data sets and to derive a new quality model for the data.

There is a trade-off between domain optimized solutions for the given data and an approach that can be universally used on other domains and similar tasks. Our general approach for matching of different data sets aims to minimize manual labor. Therefore, we focused on developing a classification technique that can be trained on a geographically small coherent region and yields proper results in geographically distant regions. We relied on established visualization components for analyzing and improving textual and geospatial matching. For example, 2D point and line plots, color coding, 3D scatterplots, and visual interaction and navigation techniques were employed. Where needed, specialized components were added, e.g., a hyperplane visualization of classifiers.

## 2 Related Work

Visual exploration or visual analytics [13] still is an emerging topic. It is used to answer questions that are hardly be solved by traditional methods. In our case, the object of the visual analysis is the behavior of the classifier and the properties of the matching data.

Manning and Schütze [12] present a good introduction to machine learning approaches, like decision trees, that are used in our work. Garg et al. [7] explain the successful combination of machine learning approaches and visual analytics; similarly to us they use scatterplots, with focus on navigation in high-dimensional spaces, but do not deal with geospatial data. Robnik-Sikonja and Kononenko [14] use scatterplots to compare the behavior of different classification methods on artificial data sets. Chen [2] uses visual analytics to select features with high information gain to develop a classifier that is similar to our feature selection.

Previous work on matching of geospatial data includes the work of Sehgal et al. [15]. They match data sets of Afghanistan from two different providers, using three different string similarity measures and physical distance. First they consider each feature independently and compare different threshold values for the corresponding feature. In the last step they combine the features and

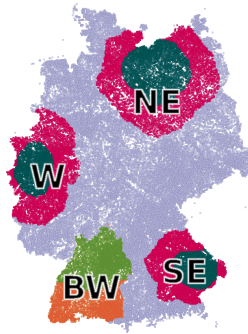


Figure 1: The area on the lower left side (corresponding to the German state Baden-Württemberg) is used as development set. It is further split into a training (north BW) and a test set (south BW). The other three areas (W, NE, and SE) are only used for the final evaluation.

learn a function to weight the different features. The main difference to our work is that they do not use any visual analysis, and we also have a richer feature set. A follow-up publication [9] describes, a graphical tool for entity resolution. The tool provides many configuration options for user-driven interactive semi-automatic matching by filtering the matching candidates, but it does not directly visualize all matching candidates to evaluate the classification.

## 3 Application Background

### 3.1 Requirements

Visual analytics should assist the development of classifiers in several tasks. First, we need visual exploration of data to get a better understanding of their characteristics. This leads to expert knowledge that can help to design features. With the visualization of the high-dimensional feature space the impact of each feature can be analyzed and optimized. Finally, the performance of the classifier should be evaluated by visual tools.

### 3.2 Data Sources

We simulate the data integration with two data sets from different providers. One data set is commercial and provided by NAVTEQ, which is one of the two leading data providers for navigation systems.

The other data set is freely available (Creative Commons License) and is managed by the Geonames project following the idea of Wikipedia. Each user has the opportunity to add, modify, and delete data. The disadvantage of such a public collaborative resource is that it is not clear how to define a homogeneous quality model and to check the consistency of the data set. Another disadvantage of the Geonames data set is that it is based on free but inexact data sources. For example, the coordinates of many locations contain only hours, minutes and no seconds in the sexagesimal notation. This will result in heavily rastered locations. Therefore, this is a challenge and stress test for our aim of finding the right corresponding data pairs. From both providers we only use the data layers that represent villages and quarters. This raises also the complexity of the matching problem because these areas constitute the lowest level of administrative area and are not well defined in many cases.

### 3.3 Classifier

Two village objects are a match if both refer to the same real world object. For making the matching decision, we use the geospatial positions and the strings of the names of the two objects. Initial experiments showed that a *search space* of 10 kilometers in the environment is sufficient to find the match for any item. For the allowed deviation of the names and positions no commonly usable rules can be defined. But in most cases the decision can be made by a human judge with high inter-judge reliability. In some marginal cases additional sources, like the homepage of the village or an online encyclopedia, must be considered to make the right decision.

Our matching component is implemented as a binary classifier that decides if a pair of village objects from the two data providers refer to the same location. We use a supervised classifier that is trained on a manually labeled training set.

### 3.4 Data Selection

Figure 1 shows all labeled sets for our experiments. We use data sets from four distinct regions of Germany to compensate for differences in their regional properties. All regions are chosen by a snowball selection process, which is used in many machine learning tasks [16] as well. It makes sure that each

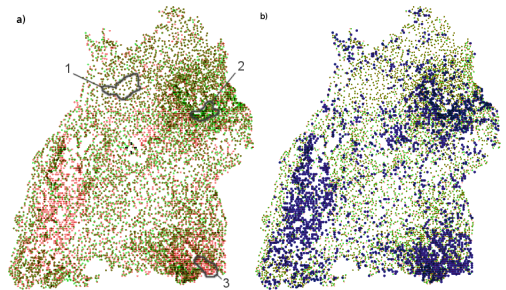


Figure 2: a) Map showing the geo-position of data entries for Baden-Württemberg from the data providers NAVTEQ and Geonames, colored green and red respectively. b) Highlighted data items (large blue points) have no correspondence.

of the four sample regions is well-connected as opposed to a set of disconnected points lacking the local information we need for matching. Further, each region is split into two parts: one for training/learning and the other for evaluation. In the development phase of our work we considered only the south-west region (which is equal to the German state Baden-Württemberg). The other three regions are only used for final evaluation tasks.

## 4 Classifier

### 4.1 Development Set

The goal of the classifier is to find a one-to-one correspondence between the NAVTEQ and Geonames data items. Figure 2 a) shows the distribution of the 6497 NAVTEQ (green) and the 7904 Geonames (red) data items in Baden-Württemberg. Each data item is drawn semitransparent with additive blending enabled. Based on the density and proximity of data points in region 1 (in top left part of Figure 2 a)) a good correspondence can be expected. In region 2 the number of NAVTEQ data items is larger whereas in region 3 the number of Geonames data items is larger. This results in many data items without correspondence in the respective other database.

### 4.2 Annotation Process

For our supervised approach, we need annotated data for training and evaluation. We have developed a tool to assist the human annotator by suggesting

possible matches. To simplify user interaction the tool makes a suggestion for possible matching pairs by ordering all pairs with a simple heuristic: similarity of the name.

Our data sets were annotated by two annotators. Although inter-annotator agreement was good [5] (see Table 1, where ( $\kappa > 0.8$ ) is good and ( $0.8 > \kappa > 0.6$ ) is satisfactory) the subsequent visual exploration highlighted errors in the annotation. A matching candidate was defined as a match if both annotators annotated it as a match. The

|          | BW   | SE   | W    | NE   |
|----------|------|------|------|------|
| $\kappa$ | 0.92 | 0.89 | 0.77 | 0.84 |

Table 1: Kappa values for annotated regions BW (Baden-Württemberg), SE (south-east Germany), W (west), and NE (north-east).

annotation process resulted in 5682 corresponding items. The remaining 815 NAVTEQ and 2222 Geonames items are highlighted in Figure 2 b): the non-corresponding items are concentrated in regions with high density differences (compare Figure 2 a)). The correspondences derived by the annotation can be represented as lines as shown in Figure 3. Corresponding items that are close together result in very short lines, not prominent in the image. Correspondences with large geospatial differences result in long lines, which are immediately visible and can be further examined. A long line does not automatically point to annotation errors. In some cases the quality of the Geonames data is bad, because it can be edited by everyone.

Annotation by humans is not free of errors. Figure 3 also shows annotation errors, found automatically, since each item can have at most one corresponding item by definition.

### 4.3 Feature Design

Our feature set was optimized on the development set of Baden-Württemberg in several iterations. In each iteration, we defined features that covered matching candidates that were not handled correctly in the previous iteration.

The spatial distance between the source and destination objects is represented in the ( $\logDist: \log_{10}(distance)$ ) feature.

The next set of features we would like to introduce is the similarity between names. Our experi-

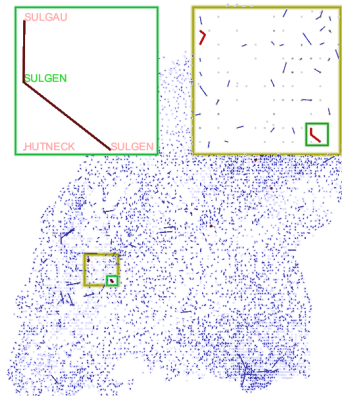


Figure 3: Corresponding NAVTEQ and Geonames data items connected with lines. Automatically found annotation errors are colored red.

ence showed that it is not possible to store all information in one feature. We started with *sim*: Trigram similarity, which is based on a trigram representation (Stuttgart -> { $\_S$ ,  $\_St$ ,  $Stu$ , ...,  $art$ ,  $rt$ ,  $t\_$ }). The similarity score of two names, a variant of the Jaccard index, is calculated by counting all equal trigrams and finally dividing them by the number of trigrams.

### 4.4 Decision Tree Classifier

We use a J48 decision tree with pruning since its classification decisions can be analyzed and understood more easily than those of many other classifiers. For each matching candidate, which is a pair of an object from Geonames and NAVTEQ, a feature vector is calculated, consisting of the above described features.

To compare the progress of the development, some metric to measure the performance is obligatory.

The classifier is trained to derive the same result as obtained by the annotation process. The classifier is not able to always derive correct results. The classifier results can be categorized by the following well known categories: positive (TP) – classifier finds a *correspondence* between two *corresponding* items; negative (TN) – classifier finds *no correspondence* between two *non-corresponding* items; false positive (FP) – classifier finds a *correspondence* between two *non-corresponding* items; and false neg-

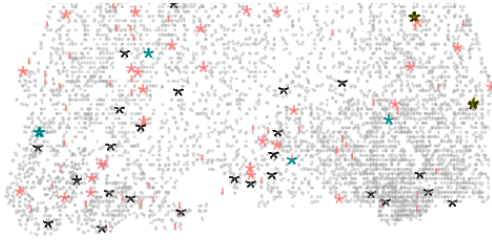


Figure 4: Star glyphs showing errors of the classifier. Each point of the star denotes an error in the respective iteration. Color coding of error types: red corresponds to false negative, black: classifier finds a correspondence between two non-corresponding items that *both* have a *correspondence* to other items; cyan: classifier finds a correspondence between two non-corresponding items that *both* have *no correspondences* to other items, and yellow: classifier finds a correspondence between two non-corresponding items where the *Geonames* data item has a *correspondence* to another item but the *NAVTEQ* data item has no correspondence.

ative (FN) – classifier finds *no correspondence* between two *corresponding* items.

#### 4.5 Visual-Aided Classifier Development

The map-based visualization of the first classification results showed that the basic algorithm is not sufficient because nearby classifications have an impact on each other. To overcome this issue an iterative algorithm [1, 11, 8] is applied. In each iteration the classification result of the previous iteration is taken as input. In the *bootstrap step* an initial classifier is trained on the training set. This classifier is then applied to the training set; the classifier results are appended to the feature vector, which is the input for the second classifier. This process can be applied iteratively. Two additional features model the previous assignments. The feature *preScore* values the score of the previous iteration. The more important new feature is the rank value *rank*. The ranking is built over the scores of matching candidates that include the same Geonames object.

To analyze the performance of a classifier, false classifications are visualized with the error category being color coded. To visualize the results of each iteration, we draw a star-shaped glyph whose points denote errors in up to five consecutive iterations.



Figure 5: Line connecting “EFRINGEN” and “EGRINGEN” showing a classifier error. On the right side of the star glyph, training set flag and classifier score are displayed.

Figure 4 shows a classifier with two iterations that uses the *sim* and *logDist* features in the first iteration and additionally the *preScore* and *rank* features in the collaborative second iteration. In the first iteration many correspondences are missing as can be seen by the amount of the red star glyphs where the first point of the star is present. In the following iteration many missing correspondences are found (red glyphs where *just* the first point of the star is present); but false positive correspondences are introduced (glyphs where the first point of the star is missing). This shows that the features are not descriptive enough to derive a proper classification.

When zooming in, a line connecting the two items is drawn with the classifier score and whether the match was part of the training set. This visualization presents all information necessary for diagnosing what went wrong in a small local region in an intuitive way. The design of additional features for improved accuracy has been greatly facilitated by this visualization. Certain names in the Geonames database fall exactly to the same position as can be seen in Figure 5 where the names of the regions are written next to the geo-position.

#### 4.6 High-Dimensional Feature Space

After the first explorations we saw the need for more similarity metrics.

Now we describe 5 of the 8 string similarity metrics we used in our system. *levenshtein*: Levenshtein distance between the two names. The boolean feature *partof*: Part-of relation first splits names into more tokens if they contain separation characters like parentheses, hyphens, and slashes and then returns 1 if one of the tokens is a substring of the name in the other data set and 0 otherwise. Sometimes names are supplemented by additional expressions. In Germany, spa towns start

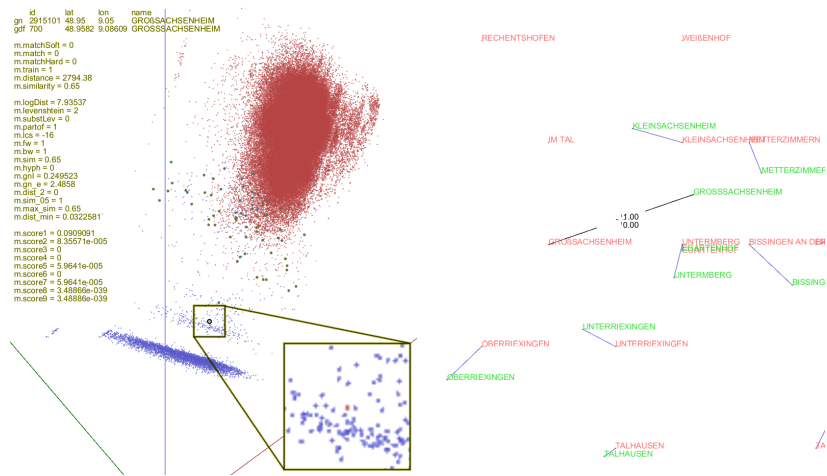


Figure 6: Linked views. The scatterplot on the left showing matching candidates. True positives, true negatives, false positives, and false negatives colored blue, red, green, and yellow respectively. The map on the right shows the position of the selected item.

with the expression *Bad*. For some spa towns, a variant without *Bad* is used, e.g., “Urach” instead of “Bad Urach”. In the same way additional prepositional phrases containing spatial information about a river (“am Neckar”, compare “upon Tyne”) can be added to names. As in the above case, these specifications are often used optionally. Therefore, we defined two special similarity measurements, *fw* and *bw*, that compute the length of the longest common prefix or suffix divided by the length of the shorter name. *hyph* is true iff one of the names includes a hyphen or a slash.

The density analyses and the classification errors of the previous features would call for features that represent the geospatial surrounding of the matching candidates. We implemented 6 features belonging to this class. As an example we describe one: the *sim\_05* feature counts other possible candidates in the vicinity that have *sim* value higher than 0.5.

The errors introduced during the classification can be divided in two classes: *systematic errors*, which are likely to be learned by the classifier, and *non-systematic errors*, which are not learned. Non-systematic errors can be detected more easily than systematic ones e.g. by examining the “false” classified items with the technique presented above.

Each matching candidate is represented as a feature vector. To detect the errors that are learned by

the classifier we need to examine the feature vectors used by the classifier. The feature vectors used by the classifier are high-dimensional, one dimension for the distance, 8 dimensions for “name distance”, and 6 dimensions for the geospatial surrounding matching candidates. We normalize the feature space to unit size and map them to 3-space with a modified FastMap algorithm. FastMap [6] maps points from  $n$ -dimensional space to  $k$ -dimensional space ( $n \leq k$ ) with the focus on preserving distances between points. We modified the FastMap algorithm to take into account the user classification according to the Supervised PCA technique [10]. The scatterplot showing the 159,973 feature vectors for Baden-Württemberg can be seen in Figure 6. Please note that on a computer screen the data items in the scatterplot can be recognized much better than on paper due to higher contrast and larger space. The selected negative match next to the positive matches is a false negative classified one that was learned by the classifier but can easily be detected in the scatterplot. By selecting a matching candidate in the scatterplot, the map on the right jumps to the selected position and allows to examine the neighboring items. The selected matching candidate “GROBSACHSENEIM”–“GROSSSACHSENEIM” was annotated as no match, but since the lowercase German letter  $\beta$

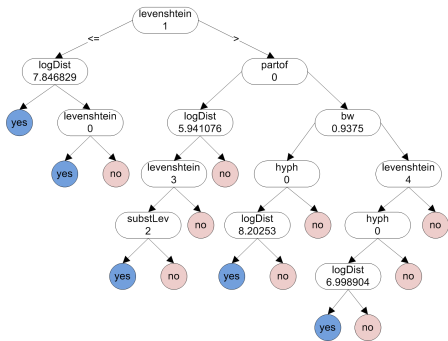


Figure 7: Decision tree used in a classifier iteration.

does not have a corresponding uppercase counterpart it is written as SS when writing uppercase. Therefore the matching candidate is a match. This is an example of the effectiveness of the visualization in identifying possible improvements to the underlying representation.

#### 4.7 Decision Tree Hyperplanes

Figure 7 shows a decision tree that is used by a classifier iteration. A decision tree takes a subspace of the feature vector space dimensions to classify each vector as positive or negative. In this case 6 of the 15 dimensions are used by the decision tree. The decision tree separates the space into two regions. We want to visualize the separating co-dimension 1 manifold to get a better understanding of the classifier. Tibshirani and Hastie [17] illustrated individual hyperplanes as lines in 2D scatterplots, but did not consider all hyperplanes defined in the decision tree. Cook et al. [3] also visualized the hyperplane defined of a Support Vector Machine by points sampled on the hyperplane. Figure 8 shows how a decision tree separates two-dimensional feature vectors. The dots represents feature vectors, blue and red color represents the classification. The decision tree divides the space along the axes and separates the differently classified feature vectors. Since the feature vectors used by our classifier are high-dimensional, we need to project them to a lower-dimensional space as indicated in Figure 8. The hyperplanes defined by the classifier are not bounded, therefore their projection would cover the whole domain. We calculate the bounding volume of the feature vectors and clip the hyperplanes with this

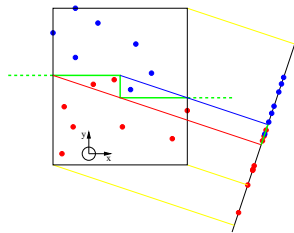


Figure 8: Two-dimensional feature vectors, classified as red and blue separated by 2-space hyperplanes (green line). In the right part of the image a projection to 1-space is shown; the separating hyperplanes within the domain are mapped to a line segment.

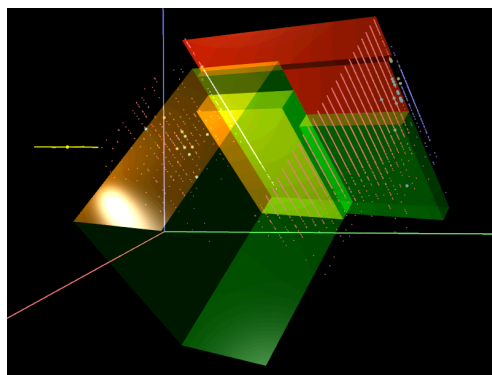


Figure 9: 6-space hyperplanes used by the classifier, lit in 3-space projection.

volume, so that the hyperplanes project to a finite volume, which is a line segment in the 1D case. On the one side of the line segment there are just blue points projected to, on the other side just red points. Within the line segment there are blue and red points that cannot be visually separated in the projection.

Figure 9 visualizes the hyperplanes of the decision tree defined in Figure 7. The feature vectors were projected orthogonally to the features *levenshtein*, *logDist*, and *partof*. In 3-space the projected hyperplanes are illuminated and rendered opaque with the projected feature vectors. Hyperplanes that divide regions based on the selected features project to planes in 3-space. Hyperplanes dividing regions based on other features project to volumes containing all feature vectors that were divided by them.

## 5 Evaluation

This section presents the overall evaluation of our approach. The introduced visual tools make the development of a classifier more convenient, due to the possibility of fast and simple data analysis, annotation assistance, and the aid in the feature design. The described visual analytics tool was developed to meet the requirements for the design of an effective classifier and greatly helped in achieving the following results, which represent a large improvement in the matching process.

| evaluation | training  | SIMPLE       | ALL          | BEST         | ITERATIVE    |
|------------|-----------|--------------|--------------|--------------|--------------|
| BW         | BW        | 52.3%        | 91.0%        | 93.0%        | 91.8%        |
| SE         | BW        | 51.1%        | 86.0%        | 86.4%        | 86.4%        |
| SE         | SE        | 72.1%        | 88.3%        | 86.7%        | 90.9%        |
| <b>W</b>   | <b>BW</b> | <b>34.6%</b> | <b>77.8%</b> | <b>81.3%</b> | <b>82.4%</b> |
| W          | W         | 62.0%        | 80.0%        | 85.4%        | 88.4%        |
| NE         | BW        | 55.0%        | 69.6%        | 66.7%        | 82.6%        |
| NE         | NE        | 48.0%        | 82.9%        | 82.0%        | 89.1%        |

Table 2: F-Scores ( $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ ) for classifiers, with precision:  $\frac{TP}{TP+FP}$  and recall:  $\frac{TP}{TP+FN}$ .

Table 2 presents the result of the final evaluation of our classifier including different feature sets. The first row contain the results for the development set and unsurprisingly, we get the highest results for the methods we optimized during the development. The following rows are of more interest. Each region is evaluated twice, first for a classifier that is trained on the development training set and second with a classifier that is trained on a close by region that has probably the same characteristics. To go into more detail we pick the bold marked row for the West (W) region that was evaluated by a classifier trained on the development set (BW). The first result column shows the result for the simple feature set that can be seen as our baseline result. The results in the other columns benefit from the previously described visual-aided methods to get more sophisticated feature sets. In the *ALL* case all revealed features are used in one feature set. The *ALL* feature set is further outperformed by the *BEST* feature set that was identified by selecting all possible subsets of the 15 features as feature sets and

evaluating them on the development set (BW). The *BEST* combination contains 5 features: *logDist*, *levenshtein*, *substLev*, *partof*, and *hyph*. The last column lists the results for the *ITERATIVE* classifier, which performs best in all cases. A further fact can be derived from the results: our advanced classifiers are less dependent on the similarity of the training and evaluation sets, compared to the *SIMPLE* classifier.

The analysis of large data sets, especially by the scatterplot shown in Figure 6, benefits from large high resolution displays. The linked view technique also requires large space, and the ability to extend the scatterplot to the whole screen was very helpful for the exploration of the data.

Usually the NLP community uses only lists and tables to analyze data. For the geospatial matching task these are not sufficient because the local interaction can hardly be represented without drawing the instances in 2D space. Also the visual density analysis assists the finding and definition of features. At last the consideration of 3D scatterplots showed obvious annotation errors that are not visible by standard NLP methods.

## 6 Conclusion and Future Work

From the visual exploration of the data several properties have emerged, which have helped to improve our classifiers. First, we have shown that annotation, although well done, can be improved by finding errors with high impact for the later classification in the scatterplot. Second, the visual analysis of the classification errors has helped to design new features to improve the classifier. Especially, false positive errors that have corresponding items motivated the use of an iterative classifier. In most cases, these classifiers perform better than non-iterative classifiers and they are robust to new domains, since the choice of the training set is not as striking as for the non-iterative classifiers.

The curse of dimensionality is a major topic when considering a high-dimensional feature space. The scatterplot visualization helps to look at the right spot but does not separate the matches from the non-matches entirely. On the other hand if a separation in a 3-space projection would exist the solution to the classification problem would be easy to find, making the problem less interesting.

In general, visual analytics tools have helped the

domain experts in NLP to develop and improve a classifier, as demonstrated by substantial improvements of the F-score results. Mostly, well known visualization component could be used, such as 2D plots, glyph plots, scatterplots, navigation, and interaction techniques. However, some specific visual mappings had to be developed, in particular, for visualizing high-dimensional feature space and decision-tree hyperplanes. Data handling (loading, conversion) also had to be adapted. Therefore, a mix of code reuse and new software parts has been useful in this application.

A future goal is to apply our approach to broad data integration tasks, like combining other data set such as Points Of Interests (POIs) of different providers. Such data could also be augmented with Wikipedia data, which also work with geospatial references.

## Acknowledgements

This project was supported in part by the DFG Collaborative Research Center/SFB 627 (NEXUS) and by the EU Coordinated Action VISMMASTER.

## References

- [1] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.
- [2] C. Chen. An information-theoretic view of visual analytics. *IEEE Computer Graphics and Applications*, 28(1):18–23, 2008.
- [3] D. Cook, D. Caragea, and V. Honavar. Visualization in classification problems. In *Proceedings in Computational Statistics*, pages 799–806. Springer-Verlag, 2004.
- [4] F. Dürr, N. Hönle, D. Nicklas, C. Becker, and K. Rothermel. Nexus—a platform for context-aware applications. In J. Roth, editor, *1. Fachgespräch Ortsbezogene Anwendungen und Dienste der GI-Fachgruppe KuVS*, pages 15–18, 2004.
- [5] B. D. Eugenio and M. Glass. The kappa statistic: a second look. *Comput. Linguist.*, 30(1):95–101, 2004.
- [6] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the ACM SIGMOD Conference*, pages 163–174, 1995.
- [7] S. Garg, J. Nam, I. Ramakrishnan, and K. Mueller. Model-driven visual analytics. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 19–26, 2008.
- [8] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 593–598, 2004.
- [9] H. Kang, V. Sehgal, and L. Getoor. GeoDDupe: a novel interface for interactive entity resolution in geospatial data. In *Proceeding of Information Visualisation*, pages 489–496, 2007.
- [10] Y. Koren and L. Carmel. Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):459–470, 2004.
- [11] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of the International Conference on Machine Learning*, pages 496–503, 2003.
- [12] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The M.I.T. Press, 1999.
- [13] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 109–116, 2004.
- [14] M. Robnik-Sikonja and I. Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008.
- [15] V. Sehgal, L. Getoor, and P. Viechnicki. Entity resolution in geospatial data integration. In *ACM GIS*, pages 83–90, 2006.
- [16] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. Technical Report CS-TR-4905, University of Maryland, College Park, 2008.
- [17] R. Tibshirani and T. Hastie. Margin trees for high-dimensional classification. *J. Mach. Learn. Res.*, 8:637–652, 2007.