	unannotated instances (328)				total instances (1484)	
rnd.	aligned	corr.	prec.	rec.	prec.	rec.
1	226	196	86.7%	59.2%	84.7%	81.6%
2	261	248	95.0%	74.9%	93.2%	91.0%
3	271	265	97.8%	80.6%	95.1%	93.8%
4	271	265	97.8%	80.6%	95.1%	93.8%

Table 1: Total learned instances

rnd.	unannot.	recog.	corr.	prec.	rec.	terms
1	331	225	196	86.7%	59.2%	262
2	118	34	32	94.1%	27.1%	29
3	79	16	16	100.0%	20.3%	4
4	63	0	0	100.0%	0%	0

Table 2: Incrementally recognized instances and learned terms

0.62, and 0.61 respectively. Since the acceptance threshold for new items is 50%, all the three locations are added to the gazetteer.

We repeat the process for several websites and show how AMBER identifies new locations with increasing confidence as the number of analyzed websites grows. We then leave AMBER to run over 250 result pages from 150 sites of the UK real estate domain, in a configuration for fully automated learning, i.e., $g = l = u = 50\%$, and we visualize the results on sample pages.

Starting with the sparse gazetteer (i.e., 25% of the full gazetteer), AMBER performs four learning iterations, before it saturates, as it does not learn any new terms. Table 1 shows the outcome of each of the four rounds. Using the incomplete gazetteer, we initially fail to annotate 328 out of 1484 attribute instances. In the first round, the gazetteer learning step identifies 226 unannotated instances. 196 of those instances are correctly identified, which yields a precision and recall of 86.7% and 59.2% of the *unannotated* instances, 84.7% and 81.6% of *all* instances. The increase in precision is stable in all the learning rounds so that, at the end of the fourth iteration, AMBER achieves a precision of 97.8% and a recall of 80.6% of the unannotated instances, and an overall precision and recall of 95.1% and 93.8%, respectively.

Table 2 shows the incremental improvements made in each round. For each round, we report the number of unannotated instances, the number of instances recognized through attribute alignment, and the number of correctly identified instances. For each round we also show the corresponding precision and recall metrics, as well as the number of new terms added to the gazetteer. Note that the number of learned terms is larger than the number of instances in round 1, as splitting them yields multiple terms. Conversely, in rounds 2 to 4, the number of terms is smaller than the number of instances, due to terms occurring in multiple instances simultaneously or already blacklisted.

We also show the behavior of AMBER with different settings for the threshold g. In particular, increasing the value of g (i.e., the support for the discovered attributes) leads to higher precision of the learned terms at the cost of lower recall. The learning algorithm also converges faster for higher values of g.

Figure 4 illustrates our evaluation of AMBER on the real-estate domain. We evaluate AMBER on 150 UK real-estate web sites, randomly selected among 2810 web sites named in the yellow pages. For each site, we submit its main form with a fixed sequence of fillings to obtain one, or if possible, two result pages with at least two result records and compare AMBER's results with a manually annotated gold standard. Using a full gazetteer, AMBER extracts data area, records, price, detailed page link, location, legal status,

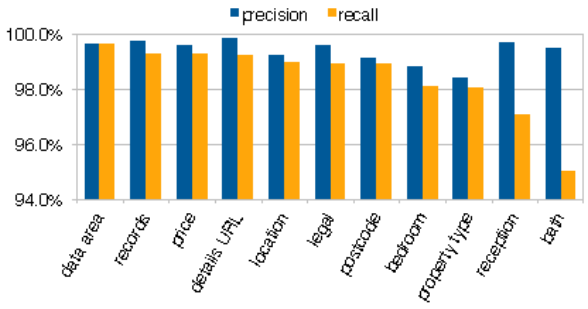

Figure 4: AMBER Evaluation on Real-Estate Domain

postcode and bedroom number with more than 98% precision and recall. For less regular attributes such as property type, reception number and bathroom number, precision remains at 98%, but recall drops to 94%. The result of our evaluation proves that AMBER is able to generate human-quality examples for any web site in a given domain.

4. REFERENCES

[1] V. Crescenzi and G. Mecca. Automatic Information Extraction from Large Websites. *Journal of the ACM*, 51(5):731–779, 2004.

[2] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters. *Text Processing with GATE (Version 6)*. The University of Sheffield, Department of Computer Science, 2011.

[3] N. N. Dalvi, P. Bohannon, and F. Sha. Robust web extraction: an approach based on a probabilistic tree-edit model. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 335–348, 2009.

[4] N. N. Dalvi, R. Kumar, and M. A. Soliman. Automatic wrappers for large scale web extraction. *The Proceedings of the VLDB Endowment*, 4(4):219–230, 2011.

[5] I. Muslea, S. Minton, and C. A. Knoblock. Hierarchical Wrapper Induction for Semistructrued Information Systems. *Autonomous Agents and Multi-Agent Systems*, 4:93–114, 2001.

[6] P. Senellart, A. Mittal, D. Muschick, R. Gilleron, and M. Tommasi. Automatic wrapper induction from hidden-web sources with domain knowledge. In *Proc. of WIDM*, pages 9–16, 2008.

[7] K. Simon and G. Lausen. ViPER: Augmenting Automatic Information Extraction with visual Perceptions. In *Proc.* 14^{th} *ACM Conference on Information and Knowledge Management*, pages 381–388, 2005.

[8] W. Su, J. Wang, and F. H. Lochovsky. ODE: Ontology-Assisted Data Extraction. *ACM Transactions on Database Systems*, 34(2), 2009.

[9] Y. Zhai and B. Liu. Structured Data Extraction from the Web Based on Partial Tree Alignment. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1614–1628, 2006.