Instant Espresso: Interactive Analysis of Relationships in Knowledge Graphs

Stephan Seufert
Max Planck Institute for Informatics
Germany
sseufert@mpi-inf.mpg.de

Patrick Ernst
Max Planck Institute for Informatics
Germany
pernst@mpi-inf.mpg.de

Srikanta J. Bedathur
IBM Research
India
sbedathur@in.ibm.com

Sarath Kumar Kondreddi
Max Planck Institute for Informatics
Germany
skondred@mpi-inf.mpg.de

Klaus Berberich
Max Planck Institute for Informatics
Germany
kberberi@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Germany
weikum@mpi-inf.mpg.de

ABSTRACT

We demonstrate InstantEspresso, a system to explain the relationship between two sets of entities in knowledge graphs. InstantEspresso answers questions of the form «*Which European politicians are related to politicians in the United States, and how?*» or «*How can one summarize the relationship between China and countries from the Middle East?*» Each question is specified by two sets of query entities. These sets (e. g. *European politicians* or *United States politicians*) can be determined by an initial graph query over a knowledge graph capturing relationships between real-world entities. InstantEspresso analyzes the (indirect) relationships that connect entities from both sets and provides a user-friendly explanation of the answer in the form of concise subgraphs. These so-called relatedness cores correspond to important event complexes involving entities from the two sets. Our system provides a user interface for the specification of entity sets and displays a visually appealing visualization of the extracted subgraph to the user. The demonstrated system can be used to provide background information on the current state-of-affairs between real-world entities such as politicians, organizations, and the like, e. g. to a journalist preparing an article involving the entities of interest. InstantEspresso is available for an online demonstration at the URL `http://espresso.mpi-inf.mpg.de/`.

1. INTRODUCTION

Knowledge graphs such as the Google Knowledge Graph[1], the Facebook graph, and the web of Linked Open Data – derived from knowledge bases such as Freebase [2], YAGO2 [8], and DBPedia [1] – are used in many applications to improve result quality and user experience in important applications such as search, recommendation, and analytics. Knowledge graphs consist of a set of semantically typed entities (e. g. artists, politicians, organizations, etc.) modeled as vertices, and labeled edges (relationships) connecting entity pairs.

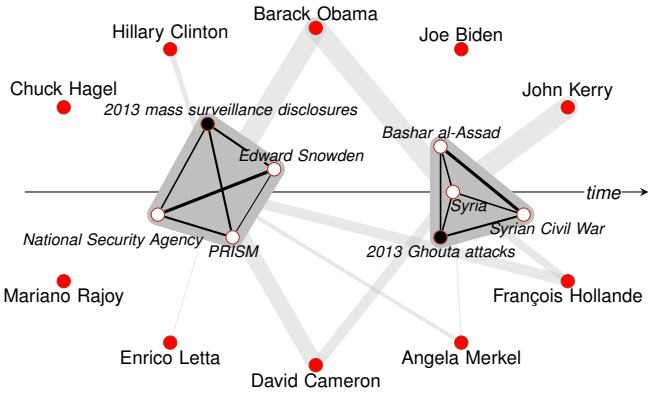

Figure 1: Summarization of the relationship between politicians from US and Europe

The InstantEspresso system demonstrated in this paper analyzes and summarizes the relationships between two sets of vertices in knowledge graphs in order to gain insight into the structure and dynamics of the underlying interactions. More specifically, we are given a knowledge graph $G = (V, E, \lambda, \mathcal{L})$ with vertices (entities) V, edges (relationships) $E \subseteq V^2$, a labeling function $\lambda : V \cup E \rightarrow \mathcal{L}$ and a set of labels $\mathcal{L}$, together with two user-specified sets of entities $Q_1, Q_2 \subseteq V$, constituting the query. InstantEspresso then provides the user with a summarization of the interactions between the entities in the query sets Q_1 and Q_2. Each summarization takes the form of a coherent, dense subgraph that is highly relevant to both entity sets. These informative graphs, called *relatedness cores*, correspond to important thematic complexes that have shaped the relationship between the entity sets.

Consider the case of a journalist interested in summarizing the political state-of-affairs between the United States and European countries in the year 2013. In this setting, the journalist would specify the query set $Q1$ as current politicians from the United

[1] `googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html`

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890528 .

States, e. g. *Barack Obama*, *John Kerry*, etc. and query set $Q2$ as the current heads of state in Europe, i. e. *David Cameron*, *Angela Merkel*, *François Hollande*, etc. In addition, the user could declare the time interval of interest, for example the year 2013. The relatedness cores computed by InstantEspresso are subgraphs that correspond to *key events* that played an important role in the relationship and occurred within the specified time interval. A desired output is shown in Figure 1. Here, the graphical relationship summarization displays the degree of involvement of each query entity in the event, together with key entities that serve as event descriptors. In the example, the events extracted for the query are the *2013 mass surveillance disclosures* – displayed as relatedness core (event complex) involving the key entities *Edward Snowden*, *National Security Agency*, etc.– and the *2013 Ghouta attacks* – described by the key entities *Syrian Civil War*, *Bashar Al-Assad*, etc. This graphical representation of relationships provides an easy-to-grasp, yet comprehensive summarization, that can be used by the journalist to quickly gain an overview over the current state-of-affairs, which in turn provides the background information of a new article.

The novel contribution of InstantEspresso lies in its ability to characterize the relationship between two *sets* of intensionally specified entities rather than between two entities [3, 4] or among the entities in a single set [11, 9]. In addition, InstantEspresso relies only on the structure of the knowledge graph and thus avoids the problem of sparsity, which limits the applicability of pattern-based approaches such as [4].

The rest of this paper is structured as follows: in Section 2, we describe the knowledge graph used to compute the results and sketch the algorithm used to compute relatedness cores. In Section 3 we discuss use cases for the demonstrated system. In Section 4 we overview the demo setup and describe how the audience can interact with the InstantEspresso system.

2. APPROACH

Espresso Knowledge Graph

InstantEspresso internally relies on the *Espresso Knowledge Graph*, $G = (V, E, \lambda, \mathcal{L})$, a graph-structured knowledge base derived from YAGO2, Freebase, as well as other, additional data sources. Two entities $(v_1, v_2) \in V^2$ in the graph are connected via an undirected edge, if the corresponding Wikipedia article pages describing the entities contain an inter-wiki link in either direction. Every edge is assigned the relationship label *hasWikipediaLink* as well as additional labels, corresponding to the relation name for each fact contained in YAGO2 between the respective entities. The resulting knowledge graph contains 4.5 million entities and 60 million relationships.

We enrich this knowledge graph by integrating several additional data sources: we incorporate edge weights signifying the relatedness between entities, derived from structural properties (inlink overlap) [10], textual descriptions of the entities based on the Wikipedia article text [7], and co-occurrence in the ClueWeb12 corpus [5]. Further, the Espresso Knowledge Graph contains semantic types associated with the entities, derived from YAGO2 (Wikipedia categories, WordNet classes) and Freebase (types). Finally, we incorporate the popularity of individual entities over time, extracted from the page view statistics of the respective Wikipedia pages[2]. InstantEspresso relies on the identification of *real-world events* highly relevant to the query entities. In order to increase both recall and precision regarding the identification of entities of type event, we have trained a linear SVM classifier to assign the type *event* to entities, based on the title and textual description of the entity. The Espresso Knowledge Graph will be made publicly available for further studies on relationship analysis at time of publication.

[2] `http://dumps.wikimedia.org/other/pagecounts-ez/`

Relatedness Core Computation

In order to compute informative relatedness cores connecting the entities in the two query sets $Q_1, Q_2 \subseteq V$, InstantEspresso performs a two-stage computation. In the first stage, a set of *candidate event entities* is extracted from the Espresso Knowledge Graph. Continuing the previous example, the *2013 mass surveillance disclosures* as well as the *2013 Ghouta attacks* are candidate event entities highly relevant to the query sets. The relatedness score of the candidate events w. r. t. the query sets is combined with additional features (such as popularity of the event) to derive a final ranking of event entities. The k highest-ranked events are then used as starting points of the individual k relatedness cores that will be presented to the user. In the second algorithmic stage, each of the k best event entities is expanded into an informative and coherent, i. e. self-contained and comprehensive, event complex – the relatedness core.

In the following, we provide an intuition on how the individual algorithmic stages are implemented.

1. The event candidates are identified by means of two random walk with restart (RWR) processes over the Espresso Knowledge Graph, one from each of the query sets. The score derived from each RWR directly captures the proximity of a vertex $v \in V$ to the corresponding query set. More precisely, without loss of generalization, we compute relatedness scores with respect to query set Q_1 as follows: first, let $M \in \mathbb{R}_{\geq 0}^{n \times n}$ denote the entity-entity relatedness matrix, containing the relatedness scores for each pair of entities directly connected in the graph. Further, let $\hat{M}$ denote the normalized (by graph Laplacian) transition matrix of the graph. For query set Q_1 we perform Jacobi power iteration over $\hat{M}$, initialized to uniform starting probabilities at the vertices $q \in Q_1$. As a result, every entity $v \in V$ of the graph is associated with a score $s_1(v)$ – obtained from the random walk's stationary distribution – signifying the relatedness of v to query set Q_1. The relatedness scores w. r. t. Q_1, Q_2 are combined with a prior probability $pr(v)$ capturing the overall importance of the respective entity, in our implementation derived from the relative amount of pageviews of the respective Wikipedia article page during the observation interval. The total score of an entity v is then given by
$$s(v) = s_1(v) \cdot s_2(v) \cdot pr(v). \quad (1)$$
2. The k highest-scoring (w. r. t. Equation 1) entities of type *event* are now expanded into informative subgraphs, the relatedness cores. In this stage of the algorithm, we first add a number of highly related entities, capturing fundamental aspects of the event complex, so-called key entities. In addition, we add entities that contribute to the explanation of the involvement of the query entities in the event complex. These so-called query context entities are highly related to both query entities as well as the central event entity, as determined by the entity-entity relatedness measure (in our implementation either based on inlink overlap or textual contents of the corresponding Wikipedia article). The result of this stage is a set of k subgraphs, each corresponding to a different thematic event complex. This second stage can take

additional constraints specified by the user, restricting the set of permitted entities in the solution.

The final output of InstantEspresso is an easy-to-comprehend visualization of the relationship summarization. More precisely, as shown in the schematic overview in Figure 1 as well as in the screenshot taken from the actual system (Figure 3), the computed relatedness cores are shown together with their connections to the individual query entities, providing a concise yet comprehensive summarization of the relationship. The cognitive load on the user is controlled by satisfying a size constraint on the number of displayed cores as well as the sizes of the individual relatedness core subgraphs.

3. USE CASES

3.1 Graphical Relationship Summarization

InstantEspresso provides a graph-based summarization of the relationship between two sets of entities, permitting non-expert users to leverage the wealth of information encoded in the underlying knowledge graph. A particularly useful application of InstantEspresso lies in the summarization of relationships between real-world entities such as countries, politicians, organizations, or athletes. During preparation of a novel article, journalists can easily research the background and interaction history of the entities of interest. As an example, for an article involving the annexation of Crimea by the Russian Federation, it is useful for the journalist to research the development of the relationship between the involving countries over recent years, i. e. – depending on the focus – between the Russian Federation and the Ukraine, or, for a wider scope between the Russian Federation and neighboring states or even between the Russian Federation and Western countries. Results returned for a query on the former scenario posed to the Espresso system include event complexes such as the *Orange Revolution* and the *Russo-Georgian War*. Through the integration of rich, structured knowledge bases (YAGO and Freebase), additional constraints can be added to the user query for a more focused retrieval. Examples include the restriction of event types (e. g. to *military conflicts*, *sports events*), geographic location, or date. In addition, users can provide keywords to constrain the possible results to entities whose description (Wikipedia article) contains the specified terms.
In the final output, edges are visualized between query entities and relatedness cores (event complexes) in a form that captures the degree of involvement. This way, it is easy to see which subsets of the query entity sets are involved in which event.

3.2 Retrieval of Relevant Documents

The purely graph-based visualization of the interaction histories provides a high-level relationship summarization. In many cases, users may require further information about the displayed event complexes. To this end, InstantEspresso provides an alignment of the displayed relatedness cores with large text corpora, allowing to directly retrieve documents (web pages and news articles) involving the relevant subsets of the query entities and the computed relatedness cores. These documents provide a textual description of the event complexes. Document retrieval is achieved by the integration of two external data sources, the ClueWeb09 corpus with annotated Freebase entities [5], as well as STICS [6], an entity-search engine over a continuously updated stream of news articles. The latter provides up-to-date news articles (currently over 1.5 million) from more than 100 feeds. As soon as a user has specified the query entities of interest, InstantEspresso first computes the graph-based relationship summarization, and subsequently queries the two corpora for documents containing entities from both sets as well as from the event complexes. For the example depicted in Figure 1, results obtained from the STICS corpus correspond to news articles involving at least one politician from each query set. The documents are ranked by a combination of traditional measures such as PageRank as well as the number of annotated relevant entities.

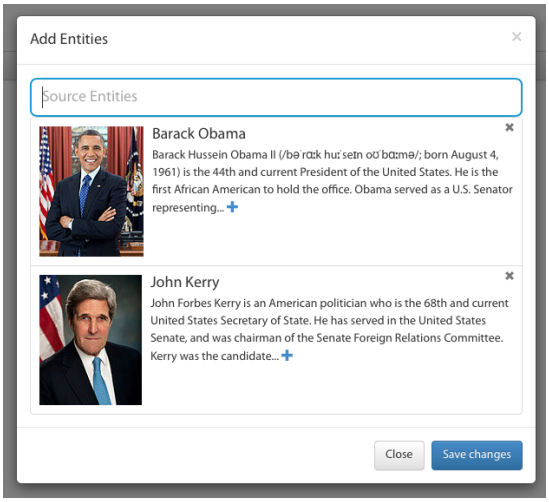

Figure 2: User Interface for Query Specification

3.3 Aggregated Relationship Summarization

In addition to providing summarizations in the form of the most important event complexes in the form of informative subgraphs, InstantEspresso also provides the user with higher-order overviews, offering an aggregated view of the interaction history between the query entities.

Relationship Heatmap. Especially for large query sets – e. g. *US politicians*, *Oil Companies*, etc. – it makes sense to first identify subsets of interrelated entities from both query sets and subsequently display the relationship summarizations for the respective subsets selected by the user. A visually appealing way to identify these subsets are heatmaps indicating the strength of interactions. Here, entities are first clustered based on their co-occurrence and then arranged as the columns and rows of a matrix representation. The matrix cells are colored based on the number of co-occurrences of the entities corresponding to row and column. Selection of a cell (submatrix) then leads to focus on the respective entity subsets for the subsequent relationship summarization by the graphical display of relatedness cores.

Topical Classification. As an additional feature, InstantEspresso aggregates the computed event complexes into a coarse-grained overview in the form of topics (e. g. sports, politics, economics, etc.). For this purpose, we have trained a multinomial classifier, assigning each article to one or more topics from a predefined set. After computing the most characteristic event complexes, InstantEspresso displays an aggregated view of the relationship based on the topics associated with the central events.

Temporal Relationship Analysis. Another insightful aggregated view is achieved by the temporal alignment of event complexes. For this purpose, InstantEspresso assigns the extracted relatedness cores to the (discretized, e. g. yearly) time axis, based on the event timespan as recorded in the Freebase knowledge base. Each time interval receives a score, based on the number of event complexes

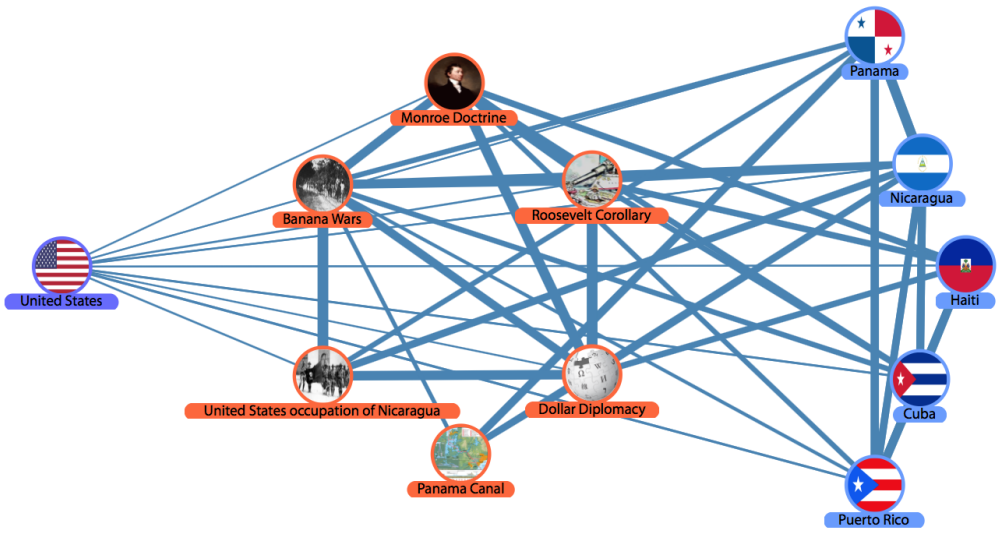

Figure 3: Relatedness Core between the United States and countries from the Americas

relevant to the interval, weighted by the degree of involvement of the individual query entities and the overall event importance (derived from structural measures such as degree centrality or PageRank, as well as popularity measured by number of pageviews). The resulting visualization gives an overview over the periods of time of highest importance to the queried relationship.

4. SYSTEM DEMONSTRATION

Hardware. InstantEspresso over the Espresso Knowledge Graph has modest hardware requirements. 8 GB of main memory are sufficient to represent the structural information of the graph used for computing the random walk scores for candidate event ranking. InstantEspresso will be demonstrated on an out-of-the-box Apple MacBook Pro, equipped with an Intel Core i7@2.3Ghz Quad Core CPU and 16 gigs of main memory.

Query Processing. The time required to answer a query is independent of the number of query entities specified by the user, since we use a random walk approach by power iteration with a fixed number of iterations. Thus, the amount of computation is constant, only the starting probabilities differ across queries. InstantEspresso provides answers to each query after roughly 20 seconds.

User Interaction. Users can interact with InstantEspresso by specifying two sets of query entities and inspecting the computed results. Entities can be specified either enumeratively or by selecting one or more predefined entity sets. The query specification interface is shown in Figure 2. In addition, users can specify a time interval of interest. InstantEspresso then first displays aggregated information, such as the heatmap and temporal overviews discussed in Section 3.3. Afterwards, the graphical representation is shown to the user (see Figure 3 for an example). Users can click on entities to see a short description of the entities. Below the graphical representation, InstantEspresso provides a list of relevant retrieved documents from ClueWeb and STICS.

5. CONCLUSION

We demonstrate InstantEspresso, a system for interactive analysis of relationships between two sets user-specified entities over a knowledge graph. Our system summarizes relationships by means of extracting important, relevant, and coherent thematic complexes corresponding to real-world events. These thematic complexes are modeled as size-constrained, dense subgraphs that are well connected with the query entities in the knowledge graph. Our system provides an interface to specify the query entities of interest, based on this selection extracts informative subgraphs from the knowledge graph, and displays the extracted results in a visually appealing graph visualization together with aggregated information and relevant news and web documents. An important target application of InstantEspresso is its use to quickly gain background information about the interaction history of entities, e. g. for a journalist preparing an article involving the query entities.

6. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, LNCS volume 4825, 2007.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD'08*.

[3] C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast Discovery of Connection Subgraphs. In *KDD'04*.

[4] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon. REX: Explaining Relationships between Entity Pairs. *PVLDB*, 5(3):241–252, 2011.

[5] E. Gabrilovich, M. Ringgaard, and A. Subramanya. FACC1: Freebase Annotation of ClueWeb Corpora, Version 1, 2013.

[6] J. Hoffart, D. Milchevski, and G. Weikum. STICS: Searching with Strings, Things, and Cats. In *SIGIR'14*.

[7] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. KORE: Keyphrase Overlap Relatedness for Entity Disambiguation. In *CIKM'12*.

[8] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 2013.

[9] G. Kasneci, S. Elbassuoni, and G. Weikum. MING: Mining Informative Entity-Relationship Subgraphs. In *CIKM'09*.

[10] D. Milne and I. H. Witten. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *WIKIAI'08*. AAAI, 2008.

[11] H. Tong and C. Faloutsos. Center-Piece Subgraphs: Problem Definition and Fast Solutions. In *KDD'06*.