CrunchCast: Matching Crunchbase and Lightcast

James Bessen*, Erich Denk*, Felix Poege[§]

*Boston University [§]Bocconi University

February 2025

Abstract

We present a novel match of companies in the Crunchbase database to online job advertisements (help wanted ads) listed in the Lightcast (formerly Burning Glass Technologies) database using a sophisticated record linkage approach, achieving approximately 90% precision and recall. These data links allow users to explore detailed hiring behavior of firms as well as aggregate ad posting behavior. The match also enables researchers to investigate the subsample of startup companies in greater detail. We discuss methods and present exemplary descriptive statistics. The data are available for research purposes.

This work was supported in part through a grant from the Alfred P. Sloan Foundation (Grant #G-2021-16968).

1. Overview

A growing body of literature investigating startup creation and performance relies on detailed startup information provided by Crunchbase. Studies explore different aspects related to (high-tech) startups, incl. the startup team, their relationship with investors, and patents (see Dalle et al., 2017 for a review). Crunchbase data has previously been used in combination with a variety of other data sources, e.g., Twitter or LinkedIn data (see Dalle et al., 2017 for a review), intellectual property data from PATSTAT (see e.g. Tarasconi & Menon, 2017), or survey data on AI startups (Bessen et al., 2022). There is also a growing literature using job vacancy data from Lightcast (formerly Burning Glass Technology). For example, studies investigate changes in labor demand in various settings, incl. changing skill demand and labor market returns (Deming & Noray, 2020), labor market concentration (Azar et al., 2020; Burya et al., 2023), or the impact of the Great Recession on skill requirements (Hershbein & Kahn, 2018). For this reason, Lightcast data coverage has been analyzed, and the vacancy data have frequently been used in combination with other data, e.g., the American Community Survey, the Current Population Survey, or Compustat (Cammeraat & Squicciarini, 2021; Hershbein & Kahn, 2018). However, very few studies have linked startup data from Crunchbase and online job postings from Lightcast.[1]

This document presents a match between Lightcast and Crunchbase that we made publicly available. We discuss the available versions of the matched and shared data, CrunchCast. The matching process relies on the Python package 'dedupe' (Gregg & Eder, 2022), and we use manually collected data for validation. We discuss the structure of CrunchCast and provide descriptive statistics for the matched data. Lastly, we describe how to use CrunchCast data when studying a subsample of startups.

We attempt to link each online job posting from Lightcast to a corresponding firm in Crunchbase. Crunchbase is a large database of corporations and includes detailed information on firm founding, financing rounds and amounts of financing, headquarters location, and markets. Of particular interest is the startup population covered by Crunchbase. We will specifically discuss firms founded after 2010, but the match covers all firms. The second data source, Lightcast, provides the near-universe of online job advertisements from 2010 to the present (until early 2021 in the data available to us). These data provide detailed information on the advertised salary, firm name, industry, occupation, required education and experience, requested skills in detail, and geographic location of the job. The main challenge when creating the match between the two data sources is related to the data. For example, firms can have multiple entries in both data sources. At the same time, firms can have multiple names and changing locations in Crunchbase. We address these challenges when preparing and matching the data. For example, we geocode startup- and online job posting locations and incorporate multiple snapshots of Crunchbase data into the matching algorithm.

We match the two data sources using the Python package 'dedupe,' which is specifically designed for record linkage. In addition, we construct validation samples to manually match a random subset of Lightcast entries to Crunchbase. This allows us to analyze how well the matching algorithm reproduces the manual choices. The match achieves approximately 90% precision and recall.

[1] Lee and Kim (2024) link startup data from Crunchbase and online job posting data from Lightcast to study when startups scale.

The publicly available matched data, CrunchCast, enables research on startup performance and growth by providing information on the hiring behavior of startups. Crunchbase provides much information about startups, while Lightcast offers detailed information about online job postings. Given that mainly more established startups post online job ads, the linked data provides the opportunity to examine their performance and growth in greater detail.

This document is outlined as follows: Section 2 discusses the data preparation of both data sources, while Section 3 describes the matching process. Section 4 provides more details on the matched and shared data CrunchCast. Section 5 provides more information on using the shared data to investigate startups.

1.1 Recommended citation

When using the data, please also cite:

Bessen, James E., Felix Poege, and Ronja Röttger. 2023. “Competing for Talent: Large Firms and Startup Growth.” SSRN Scholarly Paper. Rochester, NY. https://doi.org/10.2139/ssrn.4673494.

2. Data Preparation

We aim to link each online job posting from Lightcast to a corresponding entry in Crunchbase. Next to the matching process itself, there are two main challenges for creating the link: First, there may be multiple entries for each firm in both data sources. For example, large firms might hire at multiple locations. Second, in Crunchbase, firms may have multiple names or change their location over time. We address these challenges by cleaning key variables, e.g., firm names, geocoding locations of startups and job ads, and we incorporate multiple snapshots of Crunchbase data into the matching algorithm. This section discusses how we prepare each data source for the matching process.

2.1 Lightcast (formerly Burning Glass Technology) Data

The Lightcast (then, Burning Glass) data consists of about 180 million entries of online job ads posted between January 2010 and May 2021. We restrict the analysis sample to name, location (city, state), and industry information. Industry information (NAICS) codes are given at various levels of detail, ranging from 2-digit to 5-digit codes. At the name-location-industry level, about 24.5 million entries remain.

We clean the firm name of special characters and any potentially remaining legal titles.

2.2 Crunchbase Data

Crunchbase is a large database of corporations and includes detailed information on firm founding, financing rounds and amounts of financing, headquarters location, and markets.

As firms may have changed their names or locations over time, we incorporate multiple snapshots of Crunchbase into the matching algorithm. We include snapshots from 2013, 2019, 2020, and 2021. Data users should therefore only consider startups added to Crunchbase in 2021 or earlier.

The key firm identifier in Crunchbase is the permalink. We update older permalinks using lookups on the Crunchbase website. Changed permalinks result in forwarding to the correct new permalink, whereas deleted and obsolete permalinks result in a 'not found' error.

We collect name variants, including legal names and aliases. To avoid spurious matches, we drop overly frequent aliases. We clean all names by extracting legal titles and removing special characters.

We retain the most recent information about the founding year and firm size within a permalink, if available. Note that we do not restrict based on the founding year, as we want to present a comprehensive match between Crunchbase and Lightcast. This means that all firms listed on Crunchbase are included; therefore, the match is not based only on the subsample of startups.

Initially, we do not geographically restrict the Crunchbase entries, recognizing that foreign firms with a US presence can also post job advertisements or that firms without geographic information in Crunchbase can be valid entries, too. However, we geocode all locations (city, state, country) to obtain latitude and longitude.

We assign Crunchbase entries to NAICS industries based on the firm's business description. Crunchbase itself does not supply industry codes, and the categorizations included in Crunchbase are coarse and not unique. To assign industry codes, we use a BERT transformer model specifically trained for text similarity (Reimers & Gurevych, 2019) to encode business descriptions and the text descriptions of NAICS industry codes. We assign the best-fitting industry code at the 5-digit level based on the cosine similarity between the two encodings.

3. Matching Crunchbase and Lightcast

In this section, we describe the matching process of each online job posting from Lightcast to a corresponding entry on Crunchbase. We describe the variables used for matching and then discuss the matching process.

We use the following variables for matching:

- Firm name: The (cleaned) firm name (including aliases) from Crunchbase and the employer´s name from Lightcast. We separate legal titles and common words into separate fields.
- Location: We geocode the startup locations and the locations of the ads. We also use the country itself as a matching variable. Lightcast only covers jobs located in the US, but some foreign firms may also employ US workers.
- Industry: We use the industry listed in Lightcast and the predicted industry of a Crunchbase entry.
- Size: We use size information from Crunchbase to allow the algorithm to be more or less lenient on location factors for larger firms.
- Founding year: We use the information on the founding year and ad posting year for matching.

The matching algorithm accounts for variables that contain missing values by creating dummy variables to indicate this circumstance.

To execute the matching, we use the Python package 'dedupe,' which is specifically designed for record linkage. The first main challenge in record linkage is deciding whether any two

entries belong together. For this, Dedupe uses training data to learn a statistical model (regularized logistic regression) based on the factors mentioned above and a series of interactions between the variables. The second challenge is that it is infeasible to calculate scores for all pairs of entries. To overcome this, Dedupe uses the training data to learn efficient rules for recovering potential match candidates. For example, if Dedupe determines that having the same first word is typically in common among valid matches, this rule can be used to construct indices. Then, for each entry in Lightcast, Dedupe can rapidly retrieve the list of potential candidates from Crunchbase and calculate the required subset of scores.

Training and validation data are needed to construct the match and learn about its quality. Dedupe has a built-in mechanism for querying the user to construct training data. The library presents a pair of entries that have, according to the current state of the algorithm, a large amount of uncertainty. The user answers the algorithm's query by determining whether the entries match or whether they are distinct. In this way, the information added through training is maximized. On the other hand, the training data is not representative of the whole sample and cannot be used to learn about the match quality. For this, we construct validation samples by manually matching a random subset of Lightcast entries to Crunchbase.

3.1 Validation of the match

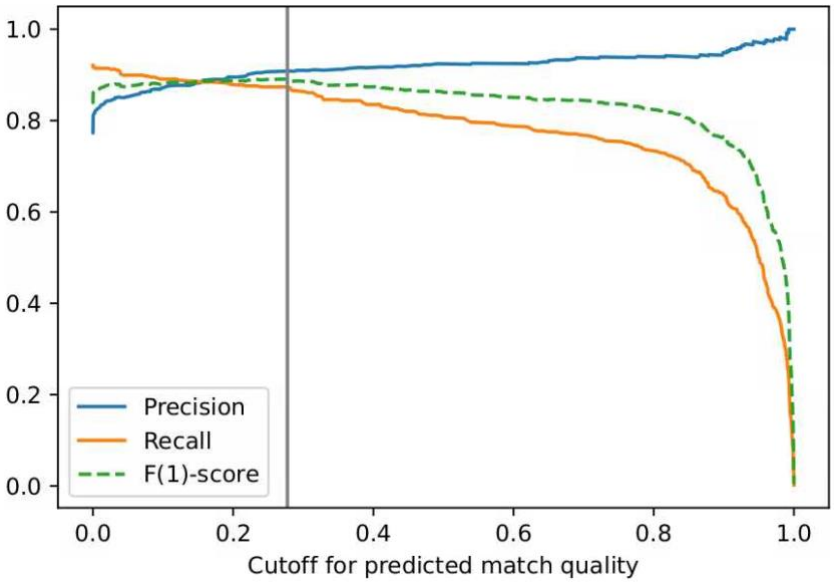
To validate the match, we randomly select 600 job advertisements from Lightcast and manually match them to the correct Crunchbase entry. We can then analyze how well the matching algorithm reproduces the manual choices.

Dedupe's matching model yields the best-fitting Crunchbase entry and a corresponding match score for each Lightcast entry. Consequently, it is necessary to select a threshold to differentiate accepted and rejected matches. Most match scores are very low or very high, but there is an intermediate range that can potentially make a difference.

For the evaluation of the match, we rely on precision and recall scores and their combination, the F1-score. Precision is the share of correctly matched entries among the entries that were matched by Dedupe, and recall is the share of correctly matched entries among all entries that should have been matched according to the manual validation data. Finally, the F1-score combines precision and recall as $F(1) = 2 * Precision * Recall \,/\, (Precision + Recall)$.

Figure 1 shows how precision and recall vary over the range of possible thresholds. As thresholds increase, precision increases, but recall declines. The maximum $F(1)$ score that can be achieved is 0.89, at precision level of 0.91 and recall of 0.87, and with a threshold of 0.277.

Figure 1: Precision-recall curve for the CrunchCast match



Note: Shows precision, recall, and F(1) score across the range of thresholds for the Dedupe match when comparing match results to manual validation results.

4. CrunchCast Data Description

4.1 Data Accessibility

The shared data is available as a link between a hash of the employer-location-year-industry combination of the ad and a Crunchbase permalink. This data release reflects the output from our model without additional modifications.

Using a hash function, we anonymize the relevant ad-level information while allowing users to recover the exact match if they possess the Lightcast data. This version omits any imputation or manual corrections and directly results from Dedupe's matching model. The data users can consider whether to apply additional appropriate imputations, such as considering the modal match valid for all entries with the same employer name.

For every hash, we include up to five match results and the corresponding score, restricted to matches with a score of at least 0.05. Before using the data, users should retain the best match result above the threshold that is included above the threshold that is included in their version of Crunchbase.

Variable	**Source and description**
bg_hash	MD5 hash of employer name, industry, city, state
Permalink	Crunchbase identifier, largely time-persistent
Score	Matching score (range 0-1). Based on validation data and the F1-score,

	we recommend a threshold of 0.277.

Hash functions are available in all common software versions. In Stata, using the 'shasum' package: shasum employer city state naics, md5(bg_hash). In Python, the combined string s should be encoded using the hashlib library with: hashlib.md5(s.encode()).hexdigest() In either case, the state is the 2-character US state code, and naics (as string variable) is the NAICS code reported in Lightcast.

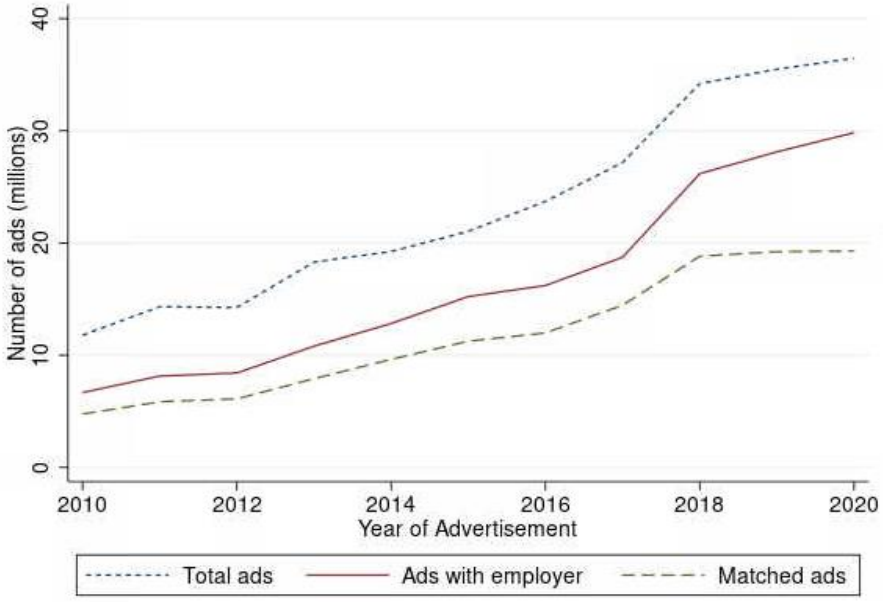
Additionally, we share a file ("startup_validation.csv"), which can be used to exclude some faulty startup entries, see Section 5.2.

4.2 Descriptive statistics

Overall, we are able to match around 80% of the ads in Lightcast to an entry in Crunchbase. Figure 2 below shows that the match rate is relatively constant over time.

When focusing on young firms, i.e., startups (founded since 2010), about 31% have at least one online job posting on Lightcast. Figure 3 shows the share of startups with at least one Lightcast ad for different employment ranges at startups. For example, among small startups with 1 to 10 employees, less than 20% of startups have at least one ad on Lightcast. In contrast, among bigger startups with 1001 to 5000 employees, about 70% of startups have at least one ad on Lightcast.[2]

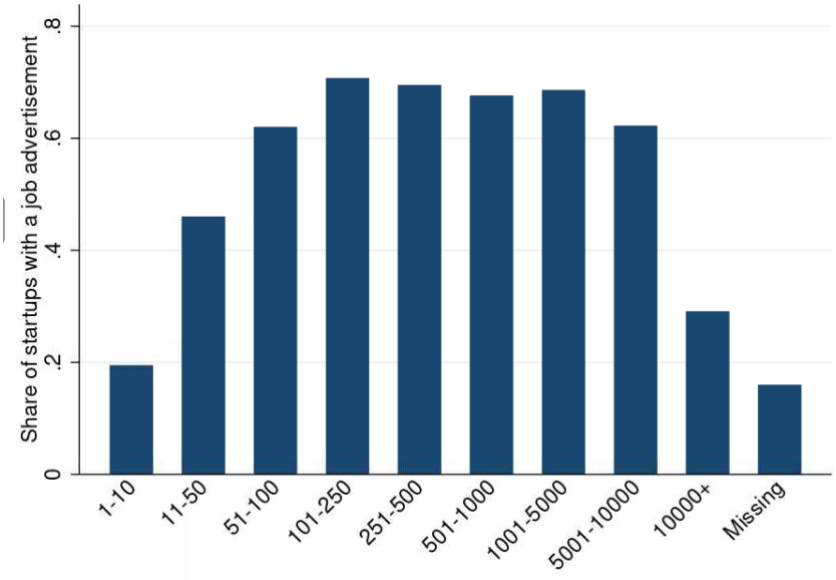
Figure 2: Descriptive statistics of the match: Matched ads



Note: Ads without employer information cannot be matched, they likely refer to staffing agencies.

[2] Note that small startups or startups without employment information are much more numerous than larger startups in Crunchbase (52% of all startups reported 1-10 employees in the 2021 snapshot, and 6.6% did not report any information, compared with less than 1% for each size class above 1000 employees). Further, a startup's first ad is on average posted three years after founding. For many startups founded in later years, ads may not have been posted yet.

Figure 3: Share of startups by Crunchbase employment ranges matched with ads



Note: Employment information is based on the 2021 version of Crunchbase. Large reported employment numbers of startups may be particularly inaccurate.

5. Analyzing startups with CrunchCast

In this section, we describe what data users of CrunchCast need to consider when using the shared data to investigate startups. We consider Crunchbase entries to be startups if they were founded after a certain point in time. We use the cutoff of 2010 to align with the data coverage of Lightcast. Additionally, due to the Crunchbase versions used for creating the match, data users should only consider startups added to Crunchbase in 2021 or earlier (see Section 2.2).

5.1 Postprocessing opportunities

In this data release, we include the direct output from the model. For empirical work, researchers can decide to enhance the match using imputation and manual correction steps.

In practice, the matching algorithm does not always assign the same Crunchbase entry to all Lightcast entries with the same employer. Inspecting these cases, researchers can decide to impute:

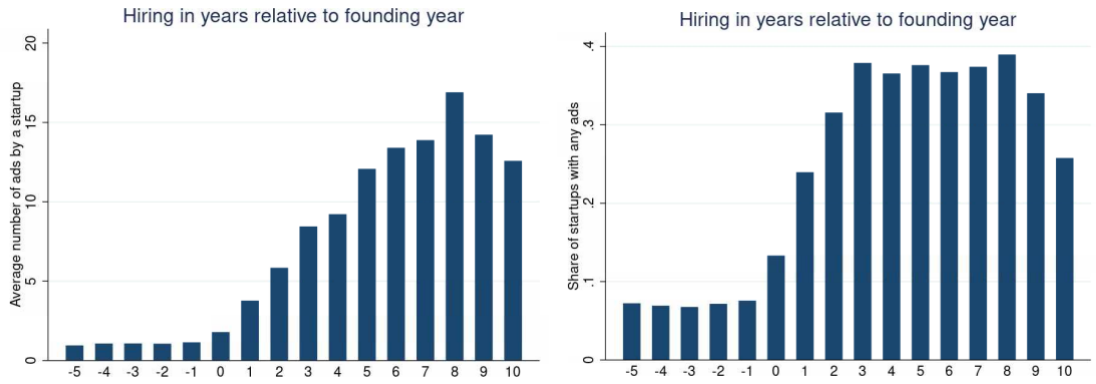
- Whenever, for some employer name, some entries are assigned to Crunchbase, and others are not, researchers can impute the unassigned ones with the most frequent permalink among the assigned ones. Such cases occurs when the match score calculated by the algorithm falls under the validation threshold for some entries but is above the threshold for others.
- Whenever, for some employer names, some entries are assigned to different Crunchbase entries, researchers can enforce a unique match by maintaining the match that was the most frequent permalink within an employer before applying the first imputation step. In practice, such cases are relatively rare.

5.2 Quality considerations

If a specific sub-population is of heightened importance, researchers can consider manual postprocessing. For example, Bessen et al. (2023) manually inspect the match for the most frequent employer names to ensure adequate match performance for the largest firms.

Additionally, it is important to note that in CrunchCast, there is a subsample of startups which is matched to job advertisements before the founding year (Figure 4). The figure shows the hiring in years relative to the founding year of startups.

Figure 4: Startup hiring/online job postings in years relative to the founding year



Based on manual inspection, the cause of the problem is typically data quality in Crunchbase. Random samples of 'startups,' i.e., firms founded since 2010 in general, are typically startups. However, when inspecting random samples of 'startups' conditional on being matched to many ads, only 50-60% of cases are startups. Many of these firms are formed as a result of M&A activity or spinouts. As their ad count can be very large, data users may arrive at faulty conclusions if not adequately corrected.

However, the number of such entries that wrongly appear as a startup and are matched to many ads is small. We provide a list that data users can apply when making use of CrunchCast, but we recommend additional testing (and encourage the sharing of additional generated data with us). We additionally recommend the following rules for excluding likely faulty entries:

- Exclude firms from a manually curated list of non-startups
- Exclude firms with ads posted before the founding year
- Exclude firms that post more than ten times the employment reported in Crunchbase (maximum of the given interval)
- Irrespective of the above rules, include firms that are a unicorn according to Crunchbase or in a manually curated list of startups.

References

Azar, J., Marinescu, I., Steinbaum, M., & Taska, B. (2020). Concentration in US labor markets: Evidence from online vacancy data. *Labour Economics*, *66*, 101886. https://doi.org/10.1016/j.labeco.2020.101886

Bessen, J., Impink, S. M., Reichensperger, L., & Seamans, R. (2022). The role of data for AI startup growth. *Research Policy*, *51*(5), 104513. https://doi.org/10.1016/j.respol.2022.104513

Burya, A., Mano, R., Timmer, Y., & Weber, A. (2023). The Wage Phillips Curve under Labor Market Power. *AEA Papers and Proceedings*, *113*, 110–113. https://doi.org/10.1257/pandp.20231007

Cammeraat, E., & Squicciarini, M. (2021). *Burning Glass Technologies' data use in policy-relevant analysis: An occupation-level assessment* (OECD Science, Technology and Industry Working Papers 2021/05; OECD Science, Technology and Industry Working Papers, Vol. 2021/05). https://doi.org/10.1787/cd75c3e7-en

Dalle, J.-M., Besten, M. den, & Menon, C. (2017). *Using Crunchbase for economic and managerial research*. OECD. https://doi.org/10.1787/6c418d60-en

Deming, D. J., & Noray, K. (2020). Earnings Dynamics, Changing Job Skills, and STEM Careers*. *The Quarterly Journal of Economics*, *135*(4), 1965–2005. https://doi.org/10.1093/qje/qjaa021

Gregg, F., & Eder, D. (2022). *Dedupe* [Computer software]. https://github.com/dedupeio/dedupe

Hershbein, B., & Kahn, L. B. (2018). Do Recessions Accelerate Routine-Biased Technological Change? Evidence from Vacancy Postings. *American Economic Review*, *108*(7), 1737–1772.

Lee, S., & Kim, J. D. (2024). When do startups scale? Large-scale evidence from job postings. *Strategic Management Journal*.

Reimers, N., & Gurevych, I. (2019, November). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. https://arxiv.org/abs/1908.10084

Tarasconi, G., & Menon, C. (2017). *Matching Crunchbase with patent data* (OECD Science, Technology and Industry Working Papers 2017/07; OECD Science, Technology and Industry Working Papers, Vol. 2017/07). https://doi.org/10.1787/15f967fa-en