Signal Discovery Strategy & Implementation Guide

Executive Summary

This document outlines the technical approach for automatically discovering business signals from various sources including predefined news sites, user-added company websites, and social media platforms. The strategy prioritizes data credibility, cost-effectiveness, and scalability.

Table of Contents

---

1. Signal Sources Overview

1.1 Source Categories

Category	Examples	Update Frequency	Credibility
Predefined News Sites	SmartCompany, Australian Tenders	Daily	High (70-80%)
Company Websites	stripe.com, airbnb.com	Weekly	Very High (90-95%)
LinkedIn	Company pages, posts	Daily	High (75-85%)
SEC Filings	8-K, 10-K, 10-Q	As filed	Highest (100%)
Press Release Sites	PR Newswire, Business Wire	Real-time	High (85-90%)

---

2. Discovery Methods by Source Type

2.1 Predefined News/Tender Sites

Sources: australiantenders.com.au, smartcompany.com.au, TechCrunch, etc.

Method 1: RSS Feeds (Recommended First)

When to use:

- Site provides RSS/Atom feeds
- Real-time updates needed
- Budget-conscious

Implementation:

```
import feedparser

def monitor_rss_feed(feed_url):
    feed = feedparser.parse(feed_url)
    for entry in feed.entries:
        signal = {
            'title': entry.title,
            'description': entry.summary,
            'url': entry.link,
            'published': entry.published,
            'source': feed_url
        }
        process_signal(signal)
```

Pros:

- Free
- Real-time updates
- Structured data
- Low bandwidth

Cons:

- Not all sites have RSS
- Limited content (summaries only)
- May miss historical data

Cost: $0

Method 2: Firecrawl API (If no RSS)

When to use:

- No RSS available
- Need full article content
- Sites with dynamic content

Implementation:

```
from firecrawl import FirecrawlApp

app = FirecrawlApp(api_key='your_key')

def crawl_news_site(base_url):
    result = app.crawl_url(
        base_url,
        params={
            'crawlerOptions': {
                'includes': ['/news/*', '/articles/*'],
                'limit': 100
            },
            'pageOptions': {
                'onlyMainContent': True
            }
        }
    )
    return result
```

Pros:

- Handles JavaScript rendering
- Clean markdown output
- Respects robots.txt
- Structured data extraction

Cons:

- Costs per page
- Rate limits
- May need page structure analysis

Cost: $0.001-0.01 per page (~$5-50/month per site)

API: https://firecrawl.dev (https://firecrawl.dev)

Method 3: Custom Scraper (Last Resort)

When to use:

- Very specific extraction needs
- Firecrawl doesn't work
- High volume justifies custom code

Pros:

- Full control
- No per-page costs
- Custom logic

Cons:

- Maintenance overhead
- Breaking changes
- IP blocking risks
- Legal considerations

Cost: Development time + infrastructure

2.2 User-Added Company Websites

Sources: User provides company website URL (e.g., stripe.com, acmecorp.com)

Recommended: Tavily Search API

When to use:

- User adds company website
- Need to discover recent signals
- Don't want to crawl entire site

Implementation:

```
from tavily import TavilyClient

client = TavilyClient(api_key='your_key')

def discover_company_signals(domain, days=90):
    # Search newsroom/blog sections
    query = f"""
        site:{domain}/newsroom OR site:{domain}/blog OR site:{domain}/press
        (product launch OR expansion OR partnership OR funding OR acquisition)
    """

    response = client.search(
        query=query,
        search_depth="advanced",
        max_results=10,
        days=days  # Last 90 days
    )

    return response['results']
```

Pros:

- No crawling infrastructure needed
- Fast (1-2 seconds)
- Time-based filtering
- Returns structured data
- Relevance scoring

Cons:

- Depends on Tavily's index
- May miss brand new content
- Costs per search
- Limited to indexed pages

Cost: $0.005 per search

Workflow:

1. User adds: acmecorp.com
2. System runs Tavily search with signal keywords
3. Extract signals from results
4. Store with source URL for data lineage
5. Re-run weekly to discover new signals

Alternative Approach:

```
# Target specific sections
sections = ['/newsroom', '/blog', '/press', '/news']
for section in sections:
    query = f"site:{domain}{section} 2024"
    results = client.search(query, max_results=5)
```

2.3 LinkedIn Company Pages

Sources: Company LinkedIn profiles and posts

Recommended: Proxycurl API

When to use:

- User provides LinkedIn URL
- Need company updates/posts
- Want employee growth data

Implementation:

```python
import requests

def get_linkedin_company_updates(linkedin_url):
    api_url = "https://nubela.co/proxycurl/api/v2/linkedin/company/updates"

    response = requests.get(api_url, params={
        'url': linkedin_url,
        'updates_count': 20
    }, headers={
        'Authorization': 'Bearer YOUR_API_KEY'
    })

    updates = response.json()

    signals = []
    for update in updates.get('updates', []):
        if is_signal_relevant(update['text']):
            signals.append({
                'title': extract_title(update['text']),
                'description': update['text'],
                'url': update['url'],
                'published': update['posted_date'],
                'engagement': update['likes'] + update['comments']
            })

    return signals
```

Pros:

- Official LinkedIn data
- Company posts/updates
- Employee count, funding info
- No scraping violations

Cons:

- Costs per request
- Rate limits
- May not capture all posts
- Requires LinkedIn URL

Cost:

- Company lookup: $0.01-0.03
- Company updates: $0.05-0.10
- ~$0.15 per company/month

API: https://nubela.co/proxycurl (https://nubela.co/proxycurl)

Alternative: Bright Data LinkedIn scraper ($500+/month for scale)

2.4 SEC Filings (Public Companies)

Sources: SEC EDGAR database

Recommended: SEC EDGAR API (Free)

When to use:

- Tracking public companies
- Need highest credibility
- Material events (acquisitions, financials)

Implementation:

```
import requests

def get_company_filings(cik, filing_type='8-K'):
    """
    8-K: Material events (acquisitions, leadership changes)
    10-K: Annual reports
    10-Q: Quarterly reports
    """
    url = f"https://data.sec.gov/submissions/CIK{cik.zfill(10)}.json"

    headers = {'User-Agent': 'YourCompany contact@example.com'}
    response = requests.get(url, headers=headers)

    data = response.json()
    recent_filings = data['filings']['recent']

    signals = []
    for i, form in enumerate(recent_filings['form']):
        if form == filing_type:
            signals.append({
                'title': f"{form} Filing",
                'filing_date': recent_filings['filingDate'][i],
                'description': recent_filings['primaryDocument'][i],
                'url': f"https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK={cik}&type={filing_type}",
                'credibility': 100
            })

    return signals
```

Pros:

- Completely free
- 100% credible (legal requirement)
- Structured data
- Real-time material events

Cons:

- Only for public companies
- Dense legal language (needs NLP)
- Limited to US companies

Cost: $0 (Free)

API: https://www.sec.gov/edgar/sec-api-documentation (https://www.sec.gov/edgar/sec-api-documentation)

2.5 Press Release Aggregators

Sources: PR Newswire, Business Wire, GlobeNewswire

Option 1: Direct API Access

PR Newswire API:

- $500-2000/month
- Real-time press releases
- Filtered by company/industry

Business Wire API:

- Enterprise pricing
- High credibility
- Broad coverage

Option 2: Tavily with Domain Filter

Implementation:

```
def search_press_releases(company_name):
    query = f'"{company_name}" (announces OR launches OR expands)'

    response = client.search(
        query=query,
        allowed_domains=[
            'prnewswire.com',
            'businesswire.com',
            'globenewswire.com'
        ],
        max_results=10,
        days=90
    )

    return response['results']
```

Pros:

- Cheaper than direct APIs
- Multiple sources
- Time filtering

Cons:

- Depends on Tavily's index
- Slight delay vs real-time

Cost: $0.005 per search vs $500+/month for direct API

3. Tool Comparison & Selection

3.1 Comprehensive Tool Matrix

Tool	Best For	Pricing	Pros	Cons	Verdict
RSS Feeds	News sites with feeds	Free	Real-time, free, structured	Not all sites have it	□ Use when available

Tool	Best For	Pricing	Pros	Cons	Verdict
Tavily	Company websites, ad-hoc searches	$0.005/search	Fast, no infrastructure, time filters	Index-dependent	□ Primary for companies
Firecrawl	News sites without RSS	$0.001-0.01/page	Smart crawling, clean output	Per-page cost	□ Secondary for news
Proxycurl	LinkedIn data	$0.05-0.15/company	Official data, no violations	LinkedIn only	□ For LinkedIn URLs
SEC EDGAR	Public companies	Free	100% credible, free	Public companies only	□ Always use when applicable
Perplexity API	AI summaries	$0.001-0.005/1K tokens	Easy implementation	Less control, AI-generated	□ Not recommended
Custom Scrapers	Special cases	Dev time + infra	Full control	Maintenance, blocking	□ Last resort
CommonCrawl	Historical data	Free but complex	Massive dataset	Stale, complex	□ Not suitable
Shovels.ai	Construction data	N/A	-	Too niche	□ Not relevant

3.2 Decision Tree

```
User adds a source
|
├─ Is it a company website?
|  |
|  ├─ YES → Use Tavily
|  |      ├─ Search: site:domain.com/newsroom OR /blog
|  |      └─ Filter: last 90 days
|  |
|  └─ NO → Continue
|
├─ Is it a LinkedIn URL?
|  |
|  ├─ YES → Use Proxycurl
|  |      └─ Get company updates feed
|  |
|  └─ NO → Continue
|
├─ Is it a public company? (has SEC filings)
|  |
|  ├─ YES → Use SEC EDGAR API
|  |      └─ Monitor 8-K filings
|  |
|  └─ NO → Continue
|
└─ Is it a predefined news site?
   |
   ├─ Check for RSS feed
   |  ├─ Has RSS → Use RSS parser
   |  └─ No RSS → Use Firecrawl
   |
   └─ Crawl news section daily
```

---

4. Recommended Architecture

4.1 System Components

```
┌─────────────────────────────────────────────────────────┐
│                  Signal Discovery Engine                 │
└─────────────────────────────────────────────────────────┘
                            │
            ┌───────────────┼───────────────┐
            │               │               │
            ▼               ▼               ▼
┌──────────────┐ ┌──────────────┐ ┌──────────────┐
│  RSS Monitor │ │ Tavily Search │ │ Proxycurl API │
│   (Cron Job) │ │   (On-demand) │ │  (Scheduled)  │
└──────────────┘ └──────────────┘ └──────────────┘
            │               │               │
            └───────────────┼───────────────┘
                            │
                            ▼
                ┌───────────────────────┐
                │  Signal Classifier    │
                │  (NLP/LLM-based)      │
                └───────────────────────┘
                            │
                            ▼
                ┌───────────────────────┐
                │  Signal Database      │
                │  (with Data Lineage)  │
                └───────────────────────┘
```

4.2 Data Flow

For User-Added Company:

```
1. User inputs: acmecorp.com + LinkedIn URL
   ↓
2. System triggers:
   - Tavily search (company website)
   - Proxycurl (LinkedIn updates)
   - SEC check (if public company)
   ↓
3. Raw results → Signal Classifier
   ↓
4. Classified signals → Database with:
   - Signal type
   - Credibility score
   - Source URL (data lineage)
   - Discovered date
   ↓
5. User sees signals in dashboard
   ↓
6. Weekly background job re-runs discovery
```

For Predefined News Sources:

```
1. Cron job runs daily (3 AM)
   ↓
2. For each source:
   - If has RSS → Parse feed
   - Else → Firecrawl latest articles
   ↓
3. Extract articles from last 24 hours
   ↓
4. Filter by user's tracked companies/keywords
   ↓
5. Classify & store matching signals
   ↓
6. Notify relevant users
```

4.3 Database Schema

```
-- Signals table
CREATE TABLE signals (
    id UUID PRIMARY KEY,
    company_id UUID REFERENCES companies(id),
    title VARCHAR(500) NOT NULL,
    description TEXT,
    signal_type VARCHAR(100), -- 'Product Launch', 'Expansion', etc.
    source_type VARCHAR(50), -- 'company_website', 'linkedin', 'news'
    source_url TEXT NOT NULL, -- For data lineage
    source_name VARCHAR(200), -- 'SmartCompany', 'Tavily', etc.
    credibility_score INTEGER, -- 0-100
    published_date TIMESTAMP,
    discovered_date TIMESTAMP DEFAULT NOW(),
    is_user_verified BOOLEAN DEFAULT FALSE,
    engagement_score INTEGER, -- For LinkedIn posts
    metadata JSONB, -- Flexible additional data
    created_at TIMESTAMP DEFAULT NOW()
);

-- Data lineage tracking
CREATE TABLE signal_lineage (
    id UUID PRIMARY KEY,
    signal_id UUID REFERENCES signals(id),
    discovery_method VARCHAR(50), -- 'tavily', 'rss', 'proxycurl'
    query_used TEXT,
    api_cost DECIMAL(10,4),
    raw_response JSONB,
    created_at TIMESTAMP DEFAULT NOW()
);

-- Source monitoring
CREATE TABLE monitored_sources (
    id UUID PRIMARY KEY,
    workspace_id UUID REFERENCES workspaces(id),
    source_url TEXT NOT NULL,
    source_type VARCHAR(50), -- 'rss', 'website', 'linkedin'
    discovery_method VARCHAR(50),
    check_frequency VARCHAR(20), -- 'daily', 'weekly'
    last_checked TIMESTAMP,
    is_active BOOLEAN DEFAULT TRUE,
    config JSONB, -- Method-specific config
    created_at TIMESTAMP DEFAULT NOW()
);
```

---

5. Implementation Phases

Phase 1: MVP (Weeks 1-2)

Goal: Basic signal discovery for user-added companies

Features:

- Tavily integration for company websites
- Manual signal review/approval
- Basic signal classification

Tools:

- Tavily API only

Deliverables:

- User can add company website
- System discovers 5-10 signals per company
- Signals displayed with source links

Cost: ~$50 setup + $0.005 per search

Phase 2: Enhanced Discovery (Weeks 3-4)

Goal: Add LinkedIn and predefined sources

Features:

- Proxycurl for LinkedIn
- RSS monitoring for 5-10 news sites
- Automated weekly re-discovery
- Credibility scoring

Tools:

- Tavily
- Proxycurl
- RSS parser (feedparser)

Deliverables:

- LinkedIn company updates
- Daily news monitoring
- Credibility badges on signals

Cost: +$100/month (Proxycurl + increased Tavily usage)

Phase 3: Advanced Features (Weeks 5-6)

Goal: SEC filings, press releases, AI classification

Features:

- SEC EDGAR monitoring (public companies)
- Firecrawl for sites without RSS
- LLM-based signal classification
- Custom signal keywords per workspace

Tools:

- All previous + SEC API + Firecrawl

Deliverables:

- 8-K filing alerts
- Smart signal categorization
- Custom keywords

Cost: +$50-100/month (Firecrawl)

Phase 4: Scale & Optimize (Weeks 7-8)

Goal: Performance, cost optimization, analytics

Features:

- Caching layer
- Batch processing
- Signal deduplication
- Analytics dashboard

Deliverables:

- 50% cost reduction via caching
- Duplicate signal detection
- Discovery metrics

---

6. Cost Analysis

6.1 Per-Company Monthly Costs

Assumptions:

- 100 companies monitored
- Weekly re-discovery
- 10 news sources monitored

Item	Unit Cost	Usage	Monthly Cost
Tavily	$0.005/search	100 companies × 4 weeks × 2 searches	$4.00
Proxycurl	$0.10/company	100 companies × 4 weeks	$40.00
Firecrawl	$0.01/page	10 sites × 30 days × 10 pages	$30.00
RSS Parsing	Free	Unlimited	$0.00
SEC EDGAR	Free	Unlimited	$0.00
Infrastructure	AWS/Railway	Server, DB, cron jobs	$50.00
LLM (Classification)	$0.002/1K tokens	1000 signals × 500 tokens	$1.00
TOTAL			**~$125/month**

Per company cost: ~$1.25/month

6.2 Scaling Costs

Companies	Tavily	Proxycurl	Firecrawl	Total/Month
100	$4	$40	$30	$125
500	$20	$200	$30	$300
1,000	$40	$400	$30	$520
5,000	$200	$2,000	$30	$2,280

Cost optimization strategies:

1. Cache Tavily results (7 days)
2. Rate limit: 1 discovery per company per week
3. Use RSS instead of Firecrawl where possible
4. Batch Proxycurl requests

With optimization:

- 500 companies: ~$150/month ($0.30/company)
- 1,000 companies: ~$250/month ($0.25/company)

6.3 Alternative: High-Volume Pricing

If scaling to 5,000+ companies:

Option 1: Custom Crawling Infrastructure

- Build in-house scrapers
- Fixed cost: $500/month (infrastructure)
- Variable: $200/month (maintenance)
- Break-even: ~600 companies

Option 2: Enterprise APIs

- Negotiate volume discounts
- Tavily Enterprise: 50% off at scale
- Proxycurl: 30% off at 10K+ requests

---

7. Data Credibility Framework

7.1 Credibility Scoring

Each signal receives a credibility score (0-100):

Source	Base Score	Factors
SEC Filing	100	Legal requirement, verified
Company Newsroom	95	Official source
Press Release Wire	90	Professional distribution
Company Blog	85	Official but less formal
LinkedIn (Company)	80	Verified account
Major News (Bloomberg, Reuters)	75	Editorial standards
Trade Publications	70	Industry-specific
LinkedIn (Individual)	60	May be opinion
Generic News	50	Verification needed

Adjustments:

- +5: Multiple sources confirm
- +10: Direct quote from executive
- -10: Older than 90 days
- -20: Contradicted by other sources

7.2 Signal Verification Workflow

```
Raw Signal Discovered
    ↓
Credibility Score Calculated
    ↓
├─ Score ≥ 80 → Auto-approve
├─ Score 60-79 → Flag for review
└─ Score < 60 → Quarantine
    ↓
User Review (optional)
    ↓
User confirms/rejects
    ↓
ML model learns from feedback
```

7.3 Source Attribution (Data Lineage)

Every signal shows:

- Primary source URL
- Discovery method
- Discovered date
- Last verified date
- Credibility score with explanation

UI Example:

```
Signal: "Acme Corp expands to APAC region"
├─ Source: acmecorp.com/newsroom/apac-expansion
├─ Discovered via: Tavily Search
├─ Credibility: 95/100 (Official company source)
├─ Published: Dec 1, 2024
└─ Last verified: Dec 10, 2024
```

---

8. Risk Mitigation

8.1 Technical Risks

Risk	Impact	Mitigation
API Rate Limits	Service disruption	Implement exponential backoff, queue system
Cost Overruns	Budget exceeded	Set monthly caps, alerting at 80%
Data Staleness	Outdated signals	Weekly re-crawls, timestamp tracking
False Positives	Poor signal quality	User feedback loop, ML improvement
API Downtime	Discovery failure	Fallback methods, retry logic
Legal (Scraping)	ToS violations	Use only approved APIs, respect robots.txt

---

8.2 Business Risks

Risk	Impact	Mitigation
Low signal volume	Poor user experience	Set expectations, manual additions
Irrelevant signals	User frustration	Better classification, user filters
Privacy concerns	Compliance issues	Only public data, clear ToS
Competitor parity	No differentiation	Focus on credibility + lineage

---

9. Success Metrics

9.1 Technical KPIs

- **Discovery Success Rate:** % of companies with ≥5 signals found
  - Target: 80% for large companies, 50% for SMBs
- **Signal Relevance:** % of signals not dismissed by users
  - Target: 70%+
- **Discovery Latency:** Time from signal published → discovered
  - Target: <24 hours for news, <7 days for company websites
- **Cost per Signal:** Total API costs / signals discovered

    - Target: <$0.10 per signal
- **Uptime:** API availability
    - Target: 99%+

9.2 Business KPIs

- **User Engagement:** % of users reviewing signals weekly
    - Target: 60%+
- **Signal Conversion:** % of signals leading to actions (outreach, etc.)
    - Target: 15%+
- **Time Saved:** Hours saved vs manual research
    - Target: 5 hours/week per user

---

10. Recommended Stack

10.1 Core Technologies

```
Signal Discovery:
  - Tavily API (primary for companies)
  - Proxycurl (LinkedIn)
  - feedparser (RSS)
  - Firecrawl (backup for websites)
  - SEC EDGAR API (public companies)

Backend:
  - Node.js/TypeScript (existing stack)
  - Bull (job queue for scheduled crawls)
  - node-cron (scheduling)

Database:
  - PostgreSQL (signals, lineage)
  - Redis (caching, rate limiting)

AI/ML:
  - OpenAI GPT-4 (signal classification)
  - Embeddings (signal similarity/deduplication)

Infrastructure:
  - Railway/AWS (hosting)
  - Sentry (error tracking)
  - DataDog (monitoring)
```

10.2 Sample Code Structure

```
/backend
  /services
    /discovery
      - tavily.service.ts
      - proxycurl.service.ts
      - rss.service.ts
      - sec.service.ts
    /classification
      - signal-classifier.ts
      - credibility-scorer.ts
    /queue
      - discovery-jobs.ts
  /models
    - signal.model.ts
    - source.model.ts
  /api
    /routes
      - signals.routes.ts
      - sources.routes.ts
```

---

11. Next Steps

Immediate Actions (This Week)

1. **Set up Tavily account**
    - Get API key
    - Test with 5 companies
    - Validate results quality
2. **RSS feed audit**
    - Check if predefined sources have RSS
    - List sources needing Firecrawl
3. **Cost estimation**
    - Calculate for current user base
    - Set budget alerts

Week 1-2

1. Implement Tavily integration
2. Build signal classification logic
3. Create data lineage tracking
4. Deploy MVP to staging

Week 3-4

1. Add Proxycurl for LinkedIn
2. Implement RSS monitoring
3. User testing & feedback
4. Production deployment

---

12. Alternatives Considered

12.1 Why Not Full Web Crawling?

Considered: Building custom crawlers with Puppeteer/Playwright

Rejected because:

- High infrastructure costs ($500+/month)
- Maintenance burden
- IP blocking risks
- Legal gray area
- 90% of pages irrelevant

When to reconsider: If scaling to 10,000+ companies and APIs become cost-prohibitive

---

12.2 Why Not All-in-One Solutions?

Considered: ZoomInfo, Apollo, Crunchbase Enterprise

Rejected because:

- Very expensive ($10K+/year)
- Limited to their data
- No customization
- Not real-time
- Vendor lock-in

When to reconsider: If building in-house isn't feasible

---

13. Conclusion & Recommendation

Final Recommendation

Phase 1 (MVP):

- Use **Tavily** for company website discovery
- Build credibility scoring
- Focus on 80% of value with 20% of effort

Phase 2 (Enhancement):

- Add **Proxycurl** for LinkedIn
- Add **RSS** for news sources
- Implement automated workflows

Phase 3 (Complete):

- Add **SEC EDGAR** for public companies
- Add **Firecrawl** for remaining sources
- Optimize costs via caching

Expected Results:

- 70-80% of companies will have 5+ signals discovered
- <$1 per company per month at scale
- 24-hour discovery latency
- 75%+ signal relevance

Total Investment:

- Development: 6-8 weeks

- Monthly operating costs: $125 (100 companies) → $250 (1000 companies)
- Per-company cost: $0.25-1.25/month

---

14. Appendix

A. API Documentation Links

- Tavily: https://docs.tavily.com (https://docs.tavily.com)
- Proxycurl: https://nubela.co/proxycurl/docs (https://nubela.co/proxycurl/docs)
- Firecrawl: https://docs.firecrawl.dev (https://docs.firecrawl.dev)
- SEC EDGAR: https://www.sec.gov/edgar/sec-api-documentation (https://www.sec.gov/edgar/sec-api-documentation)
- feedparser: https://feedparser.readthedocs.io (https://feedparser.readthedocs.io)

B. Legal Considerations

Compliant Approaches:

- Using official APIs (Tavily, Proxycurl)
- Respecting robots.txt
- Following Terms of Service
- Only accessing public data
- Attributing sources properly

Avoid:

- Scraping LinkedIn directly
- Bypassing paywalls
- High-frequency requests without rate limiting
- Storing personal data without consent

C. Contact for Tools

Tool	Sales Contact	Free Tier
Tavily	sales@tavily.com	1,000 searches/month
Proxycurl	hello@nubela.co	10 credits (~2 companies)
Firecrawl	hello@firecrawl.dev	500 pages/month

Document Version: 1.0

Last Updated: December 11, 2024

Prepared By: Signal Discovery Team

Review Status: Ready for client review